



Fleet-Level Selective Maintenance and Aircraft Scheduling

**C. Richard Cassady
Scott J. Mason
Stephen Ormon
Kellie Schneider
Chase Rainwater
Mauricio Carrasco
Jason Honeycutt**

**University of Arkansas
Department of Industrial Engineering
4207 Bell Engineering Center
Fayetteville, AR 72701**

July 2003

Interim Report for November 2002 to July 2003

20060418079

**Approved for public release;
distribution is unlimited.**

**Human Effectiveness Directorate
Warfighter Readiness Research Division
Logistics Readiness Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604**

NOTICES

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data, does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them

This report was cleared for public release by the Air Force Research Laboratory Wright Site Public Affairs Office (AFRL/WS) and is releasable to the National Technical Information Service (NTIS). It will be available to the general public, including foreign nationals.

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Federal Government agencies and their contractors registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, VA 22060-6218

DISCLAIMER

This Technical Report is published as received and has not been edited by the Air Force Research Laboratory, Human Effectiveness Directorate.

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2006-0011

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

//SIGNED//

DANIEL R. WALKER, Colonel, USAF
Chief, Warfighter Readiness Research Division
Human Effectiveness Directorate

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) July 2003		2. REPORT TYPE Interim		3. DATES COVERED (From - To) November 2002 - July 2003	
4. TITLE AND SUBTITLE Fleet-Level Selective Maintenance and Aircraft Scheduling				5a. CONTRACT NUMBER F33615-99-D-6001 DO#23	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 63231F	
				5d. PROJECT NUMBER	
6. AUTHOR(S) C. Richard Cassady, Scott J. Mason, Stephen Ormon, Kellie Schneider, Chase Rainwater, Mauricio Carrasco, Jason Honeycutt				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER 49230031	
				8. PERFORMING ORGANIZATION REPORT NUMBER MM0202	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Arkansas Dept. of Industrial Engineering 4207 Bell Engineering Center Fayetteville, AR 72701				10. SPONSOR/MONITOR'S ACRONYM(S)	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Human Effectiveness Directorate Warfighter Readiness Research Division Air Force Materiel Command Logistics Readiness Branch Wright-Patterson AFB OH 45433-7604				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-HE-WP-TR-2006-0011	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited. AFRL/WS-06-0408, 14 Feb 06					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The objective of this research is to investigate the use of a mathematical modeling methodology for integrating maintenance planning and sortie scheduling issues. First, the relevant research literature for both selective maintenance and fleet assignment is presented. Next, background research is presented, which extends a current selective maintenance model to incorporate sets of systems. Here a selective maintenance model for a set of systems that must execute a set of missions with system maintenance performed only between missions is defined. Finally, we formulate a more complex optimization model that addresses a more dynamic mission profile. Specifically, missions start and end at different times, and maintenance and scheduling decisions are made over a series of time "buckets." We consider a planning horizon such that each system in the set returns from its previous mission and begins its future mission; however, no system returns from its future before the end of the planning horizon.					
15. SUBJECT TERMS Fleet Assignment, Selective Maintenance, Aircraft Scheduling, Mathematical Modeling					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 82	19a. NAME OF RESPONSIBLE PERSON Cheryl L. Batchelor
a. REPORT UNCLASSIFIED	b. ABSTRACT UNCLASSIFIED	c. THIS PAGE UNCLASSIFIED			19b. TELEPHONE NUMBER (include area code) 937-656-4392

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE LEFT INTENTIONALLY BLANK

Contents

FIGURES.....	IV
TABLES	VI
EXECUTIVE SUMMARY	VII
1 INTRODUCTION	1
2 RESEARCH LITERATURE REVIEW.....	3
2.1 SELECTIVE MAINTENANCE RESEARCH	3
2.2 FLEET ASSIGNMENT RESEARCH.....	4
3 BACKGROUND RESEARCH.....	5
3.1 HYPOTHETICAL SET OF SYSTEMS	5
3.2 SYSTEM PERFORMANCE	6
3.3 SELECTIVE MAINTENANCE MODEL FOR A SET OF SYSTEMS	9
3.4 SOLUTION PROCEDURES.....	14
4 THE STATIC MODEL.....	18
4.1 HYPOTHETICAL SET OF SYSTEMS	18
4.2 SYSTEM PERFORMANCE	19
4.3 FORMULATION OF THE STATIC MODEL.....	22
4.4 SOLUTION PROCEDURE	26
5 THE DYNAMIC MODEL.....	32
5.1 HYPOTHETICAL SET OF SYSTEMS	32
5.2 SYSTEM PERFORMANCE	33
5.3 FORMULATION OF THE DYNAMIC MODEL	36
5.4 SOLUTION PROCEDURE	41
6 FUTURE OPPORTUNITIES.....	48
REFERENCES	49
APPENDIX A: SPREADSHEET INSTRUCTIONS FOR EVALUATING MODEL P.....	51
APPENDIX B: SPREADSHEET INSTRUCTIONS FOR EVALUATING MODEL SP.....	65
APPENDIX C: SPREADSHEET INSTRUCTIONS FOR EVALUATING MODEL DSP.....	69

Figures

FIGURE 1: EXAMPLE SET OF SYSTEMS	6
FIGURE 2: STATUS PRIOR TO MISSION	13
FIGURE 3: STATUS AFTER MAINTENANCE.....	14
FIGURE 4: PARENT CHROMOSOMES CHOSEN FOR CROSSOVER	15
FIGURE 5: CROSSOVER ROUTINE AND RESULTING CHILD CHROMOSOMES.....	15
FIGURE 6: GA LOGIC	17
FIGURE 7: EXAMPLE SET OF SYSTEMS	19
FIGURE 8: PARENT CHROMOSOMES CHOSEN FOR CROSSOVER	27
FIGURE 9: STEP ONE OF CROSSOVER ROUTINE.....	28
FIGURE 10: STEP TWO OF CROSSOVER ROUTINE	29
FIGURE 11: STATUS PRIOR TO MAINTENANCE.....	30
FIGURE 12: MISSION ASSIGNMENT AND STATUS AFTER MAINTENANCE.....	31
FIGURE 13: EXAMPLE SET OF SYSTEMS	32
FIGURE 14: PARENT CHROMOSOMES CHOSEN FOR CROSSOVER	42
FIGURE 15: STEP ONE OF CROSSOVER ROUTINE.....	43
FIGURE 16: STEP TWO OF CROSSOVER ROUTINE	44
FIGURE 17: EXAMPLE DYNAMIC MISSION PROFILE.....	45
FIGURE 18: DYNAMIC MISSION PROFILE WITH COMPONENT FAILURES	46
FIGURE 19: SOLUTION FOR NUMERICAL EXAMPLE.....	47
FIGURE A 1: MACRO NOTIFICATION	51
FIGURE A 2: INPUTS WORKSHEET	52
FIGURE A 3: UPDATED INPUTS WORKSHEET.....	53
FIGURE A 4: INPUT PARAMETERS FOR COMPONENT RELIABILITY AND NUMBER OF COMPONENTS.....	54
FIGURE A 5: INPUT PARAMETERS FOR LIMITED MAINTENANCE RESOURCES	55
FIGURE A 6: INPUTS WORKSHEET WITH EXAMPLE PARAMETER VALUES	56
FIGURE A 7: MODEL OUTPUT.....	58
FIGURE A 8: OUTPUT WORKSHEET	58
FIGURE A 9: MACRO NOTIFICATION	59
FIGURE A 10: GA SETTINGS WORKSHEET	60
FIGURE A 11: INPUT WORKSHEET WITH EXAMPLE PARAMETER INPUTS	60
FIGURE A 12: INPUTS WORKSHEET WITH EXAMPLE PARAMETERS	62

FIGURE A 13: MODEL OUTPUT	64
FIGURE B 1: MACRO NOTIFICATION.....	65
FIGURE B 2: GA SETTINGS WORKSHEET.....	65
FIGURE B 3: INPUTS WORKSHEET WITH EXAMPLE PARAMETERS.....	67
FIGURE B 4: MODEL OUTPUT	68
FIGURE C 1: MACRO NOTIFICATION.....	69
FIGURE C 2: GA SETTINGS WORKSHEET.....	70
FIGURE C 3: INPUTS WORKSHEET WITH EXAMPLE PARAMETERS.....	71
FIGURE C 4: PARAMETERS FOR DYNAMIC SCENARIO EXAMPLE	72
FIGURE C 5: INPUTS WORKSHEET WITH ALL EXAMPLE PARAMETERS.....	73
FIGURE C 6: SUGGESTED MAINTENANCE ACTIONS AND MISSION ASSIGNMENT	74

Tables

TABLE 3.1: MODEL PARAMETERS FOR EXAMPLE SET OF SYSTEMS.....	12
TABLE 3.2: FAILED COMPONENTS FOR EXAMPLE SET OF SYSTEMS	13
TABLE 3.3: OPTIMAL SELECTIVE MAINTENANCE ACTIONS.....	13
TABLE 4.1: MODEL PARAMETERS FOR EXAMPLE SET OF SYSTEMS.....	30
TABLE 5.1: MODEL PARAMETERS FOR EXAMPLE SET OF SYSTEMS.....	45
TABLE A 1: MODEL PARAMETERS FOR EXAMPLE SET OF SYSTEMS.....	55
TABLE A 2: FAILED COMPONENTS FOR EXAMPLE SET OF SYSTEMS	57
TABLE A 3: MODEL PARAMETERS FOR EXAMPLE SET OF SYSTEMS.....	61
TABLE A 4: FAILED COMPONENTS FOR EXAMPLE SET OF SYSTEMS	62
TABLE B 1: MODEL PARAMETERS FOR EXAMPLE SET OF SYSTEMS	66
TABLE B 2: FAILED COMPONENTS FOR EXAMPLE SET OF SYSTEMS.....	68
TABLE C 1: MODEL PARAMETERS FOR EXAMPLE SET OF SYSTEMS	71

Executive Summary

All industrial and military organizations rely on the effective use of repairable systems. In many cases, these organizations rely on sets of identical systems that are required to perform sets of missions. Often, sufficient maintenance resources aren't available to keep all systems within a set in top condition at all times; therefore, a maintenance manager must decide how to allocate the available resources. This allocation falls within the domain of selective maintenance. Selective maintenance is defined as the process of identifying the subset of maintenance activities to perform from a set of desired maintenance actions. Although the modeling of repairable equipment has been studied extensively, traditional studies tend to focus on a single system and ignore the mission profile of the system. For the United States Air Force (USAF), these limitations prevent current models from providing meaningful guidance relative to maintenance planning and sortie scheduling.

The objective of this project is to investigate the use of a mathematical modeling methodology for integrating maintenance planning and sortie scheduling issues. First, the relevant research literature for both selective maintenance and fleet assignment is presented. The selective maintenance literature is limited in that current models only consider a single system, and most of the fleet assignment literature is motivated by the commercial airline industry, which focuses on meeting Federal Aviation Administration (FAA) maintenance requirements.

Next, background research is presented, which extends a current selective maintenance model to incorporate sets of systems. Here a selective maintenance model for a set of systems that must execute a set of missions with system maintenance performed only between missions is defined. Originally, the scenario evaluated in the background research was done using a total enumeration strategy. Although the enumeration strategy guarantees an optimal solution, scenarios involving more than four systems result in lengthy computational times; therefore, a genetic algorithm (GA) was developed to solve larger scenarios. This background research and the development of the GA contains key concepts used in the formulation and solution procedures of both the static and dynamic optimization models that integrate aircraft assignment and selective maintenance.

The model developed in the background research does not address mission assignment, because all missions in the upcoming set are identical. In an extension to this model, we formulate a mathematical optimization model that integrates system assignment (given a relatively static mission profile) and selective maintenance decision-making. Specifically, we consider upcoming missions that are not

identical and define a parameter to quantify the “difficulty” of each upcoming mission. We develop a solution procedure for performing this optimization and study the behavior of the model using numerical examples. Because of the complexity of the optimization model, we utilize a GA to perform the required optimization.

Finally, we formulate a more complex optimization model that addresses a more dynamic mission profile. Specifically, missions start and end at different times, and maintenance and scheduling decisions are made over a series of time “buckets.” We consider a planning horizon such that each system in the set returns from its previous mission and begins its future mission; however, no system returns from its future before the end of the planning horizon. We again utilize GAs to perform the required optimization.

1 Introduction

All military organizations depend upon the reliable performance of repairable systems for the successful completion of missions. Although the use of mathematical modeling for the purpose of modeling repairable systems and designing optimal maintenance policies for these systems has received an extensive amount of attention in the literature, this research is limited in two key ways. First, studies in the literature tend to focus on a single system rather than a set of systems used by the organization. Second, these studies ignore the mission profile of the system, which prevents the modeler from considering important maintenance strategies including:

- ◆ Performing maintenance during scheduled downtime
- ◆ Delaying maintenance to execute a critical mission

For the USAF, these limitations are too severe to provide meaningful guidance relative to fleet maintenance planning. For a single aircraft, maintenance actions should be planned relative to its mission schedule, and sortie scheduling decisions should be managed with considerations for aircraft maintenance. Given that aircraft within the USAF share maintenance resources (spares, labor, etc.) and perform missions in groups, the integration of sortie scheduling and maintenance planning can become quite complex.

The objective of this project is to investigate the use of a mathematical modeling methodology for integrating maintenance planning and sortie scheduling issues. Achieving the objective of the project requires the completion of five key activities.

First, we define a hypothetical set of systems that could correspond to aircraft, subsystems within an aircraft, or subsystems within subsystems. This definition includes the number of systems, the mission profile and the constrained maintenance resources.

Second, we formulate a mathematical optimization model that integrates system assignment (given a relatively static mission profile) and selective maintenance decision-making. Note that selective maintenance refers to the process of identifying the subset of actions to perform from a set of desirable maintenance actions.

Third, we develop a solution procedure for performing this optimization and study the behavior of the model using numerical examples. Because of the complexity of the optimization model, we use a GA to perform the required optimization.

Fourth, we formulate a more complex optimization model that addresses a more dynamic mission profile.

And finally, we again utilize GAs to perform the required optimization.

The remainder of this report is organized as follows. *Section 2* summarizes the relevant research literature. *Section 3* details a description of background research critical to this project. *Section 4* formulates the optimization model for the static scenario and describes the GA-based solution approach. *Section 5* contains a similar presentation for the dynamic scenario. Finally, *Section 6* describes future opportunities.

2 Research Literature Review

This project builds upon the body of knowledge in both selective maintenance and aircraft fleet assignment; therefore, in this section, we summarize the relevant literature in both areas.

2.1 Selective Maintenance Research

Selective maintenance falls under the domain of maintenance modeling and optimization. Fortunately, the use of mathematical modeling for the purpose of modeling repairable systems and designing optimal maintenance policies for these systems has received an extensive amount of attention in the literature [8, 10, 14, 15, 18, 19, 20]. The original study in selective maintenance was performed by Rice *et al.* [16]. They define a system that must complete a series of missions where maintenance is performed only during finite breaks between missions. Due to the limited maintenance resource (in this case, time), it may not be possible to repair all failed components before the next mission. A nonlinear, discrete selective maintenance optimization model is developed that is designed to maximize system reliability for the next mission. The numbers of components to be repaired are the decision variables, and the limitation on maintenance time serves as the primary functional constraint. Due to the complexity of the model, total enumeration is the recommended solution procedure. Given that total enumeration is ineffective for large scenarios, a heuristic selective maintenance procedure is developed.

Cassady *et al.* [3, 4] extend the work of Rice *et al.* [16] in several ways. First, a more complex system is defined. Specifically, a system is comprised of independent subsystems connected in series with the individual components in each subsystem connected in some fashion. Next, the selective maintenance model is extended to consider the case where both time and cost are constrained. This leads to the development of three different selective maintenance models. These models include maximizing system reliability subject to both time and cost constraints; minimizing system repair costs subject to a time constraint and a minimum required reliability level; and minimizing total repair time subject to both cost and reliability constraints.

Cassady *et al.* [5] build upon the work of Rice *et al.* [16] in two other ways. First, system components are assumed to have Weibull life distributions. This assumption permits systems to experience an increasing failure rate (IFR) and requires monitoring of the age of components. Second, the selective maintenance model is formulated to include three maintenance actions: minimal repair of failed components, replacement of failed components, and replacement of functioning components (preventive maintenance).

Chen *et al.* [7] expand the work of Rice *et al.* [16] and Cassady *et al.* [3] by considering systems in which each component and the system may be in $K + 1$ possible states, 0, 1, ..., K . They use an optimization model to minimize the total cost of maintenance activities subject to a minimum required system reliability.

2.2 Fleet Assignment Research

Several studies in the literature address aircraft fleet assignment. Abara [1] formulates a mixed-integer programming fleet assignment model. The objective function of the model may maximize profit, minimize cost, or optimize the use of a specific type of aircraft. He assumes a certain amount of “stand-by” time for routine maintenance between arrivals and departures. Due to the computational difficulties associated with modeling fleet assignment as a mixed-integer program, Hane *et al.* [12] investigate the effect of several optimization improvement methodologies on computational time. Rushmeier *et al.* [17] improved upon the work of Abara [1] and Hane *et al.* [12] by adding “sit activity arcs,” which can be used to distinguish between aircraft on the ground. This extension could allow for a more meaningful analysis on maintenance and crew resource utilizations.

Lam [13] provides a brief history of commercial aircraft maintenance. He highlights the effects of economics, technology, and regulation of the FAA on aircraft maintenance. The different types of aircraft maintenance required by the FAA are defined and compared by the amount of labor hours, maintenance hours, and the maintenance interval hours required by an average aircraft. The actual maintenance tasks performed on an aircraft are also defined for each type of maintenance check. Barnhart *et al.* [2] develop a modeling methodology for integrating fleet assignment and aircraft routing. They develop an example using data from commercial airlines and determine that the proposed model successfully addresses both fleet assignment and aircraft routing. Feo *et al.* [11] develop a model that can be used to locate maintenance stations and to develop flight schedules that meet maintenance requirements. Specifically, the model seeks to optimize the number of required maintenance facilities to meet FAA maintenance regulations. The network model is solved using a heuristic and demonstrated using data obtained from American Airlines. The model is applied to the American Airlines Boeing 727 fleet, and the results show significant improvement over other aircraft scheduling models. Similar results are obtained when the model is applied to the Super 80 and DC-10 fleets. Clarke *et al.* [9] expand on the model developed by Hane *et al.* [12] by adding maintenance- and crew-related constraints to the fleet assignment model. The maintenance constraints address both short- and long-duration maintenance.

3 Background Research

Previously published research in the area of selective maintenance focuses on a single system. In an unpublished research effort preceding this project, Cassady and Schneider [6] formulate an optimization model to extend the work of Rice *et al.* [16] by defining a selective maintenance model for a set of systems that must perform a set of missions with system maintenance performed only between missions. This background research contains key concepts used in the formulation of both the static and dynamic optimization models that integrate aircraft assignment and selective maintenance.

3.1 Hypothetical Set of Systems

Consider a set of q independent and identical systems. Each system is comprised of m independent subsystems connected in series, and each subsystem, say subsystem j , contains n_j independent and identical copies of a constant failure rate (CFR) component connected in parallel. A graphical representation (reliability block diagram) of an example set of systems is shown in *Figure 1*. Note that each component in *Figure 1* is labeled " i, j, k " where i denotes the system number, j denotes the subsystem number and k denotes the component number.

The set of systems is required to perform sequential sets of missions that originate from and return to a common base of operation. Each set of missions contains q independent and identical missions with common start times and durations. At any time, a specific component, subsystem or system is in one of two states: functioning (1) or failed (0). Components only fail during missions, and failed components can be repaired only during the breaks between sets of missions. Note that all maintenance is performed at the base.

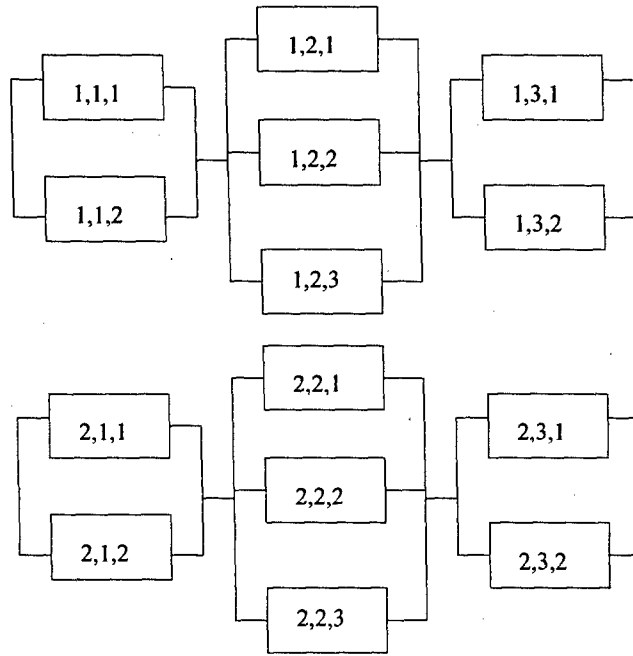


Figure 1: Example Set of Systems

3.2 System Performance

Suppose that a set of missions has just ended and all systems have returned to their base of operation and maintenance. Note that component ijk refers to component k in subsystem j of system i , and note that subsystem ij refers to subsystem j in system i . Let Y_{ijk} denote the status of component ijk , let Y_{ij} denote the status of subsystem ij , and let Y_i denote the status of system i . Since each subsystem is a parallel arrangement of its components, then

$$Y_{ij} = \prod_{k=1}^{n_j} Y_{ijk} = 1 - \prod_{k=1}^{n_j} (1 - Y_{ijk})$$

Equation 3.1

Since each system is a series arrangement of its subsystems, then

$$Y_i = \prod_{j=1}^m Y_{ij}$$

Equation 3.2

Note that if $Y_{ij} = 0$ for some $j = 1, 2, \dots, m$, then $Y_i = 0$ and system i did not successfully complete the last mission.

Prior to the next set of missions, maintenance (repair) can be performed on some or all of the failed components. Let X_{ijk} denote the status of component ijk at the start of the next mission, let X_{ij} denote the status of subsystem ij at the start of the next mission, and let X_i denote the status of system i at the start of the next mission. Note that

$$X_{ij} = \prod_{k=1}^{n_j} X_{ijk}$$

Equation 3.3

and

$$X_i = \prod_{j=1}^m X_{ij}$$

Equation 3.4

Furthermore, note that if $Y_{ijk} = 0$ and $X_{ijk} = 1$, then component ijk was repaired during the break between missions. Also, note that if $X_{ij} = 0$ for some $j = 1, 2, \dots, m$, then $X_i = 0$ and system i is unable to undertake its next mission.

Clearly, the X_{ijk} values have a direct effect on the status of components, subsystems and systems at the end of the upcoming mission. Let Y_{ijk}^+ denote the status of component ijk at the end of the next mission, let Y_{ij}^+ denote the status of subsystem ij at the end of the next mission, and let Y_i^+ denote the status of system i at the end of the next mission. Note that

$$Y_{ij}^+ = \prod_{k=1}^{n_j} Y_{ijk}^+$$

Equation 3.5

and

$$Y_i^+ = \prod_{j=1}^m Y_{ij}^+$$

Equation 3.6

Also, note that if $X_{ijk} = 0$, then $Y_{ijk}^+ = 0$. If $X_{ijk} = 1$ then Y_{ijk}^+ is a random variable. Assuming $X_i = 1$, then both Y_{ij}^+ for all $j = 1, 2, \dots, m$ and Y_i^+ are random variables. Furthermore, note that if $Y_{ij}^+ = 0$ for some $j = 1, 2, \dots, m$, then $Y_i^+ = 0$ and system i did not successfully complete the next mission.

One key factor in making maintenance decisions is the probability that each system will successfully complete its next mission. We assume that all components have constant failure rates, that is, the probability that a component successfully completes a mission depends only upon its status at the beginning of the mission and the length of the mission. Since all missions have identical length, we ignore mission length in our analysis. Let

$$r_j = \Pr(Y_{ijk}^+ = 1 | X_{ijk} = 1)$$

Equation 3.7

Note that this probability is only indexed on j because systems are identical and the components within a subsystem are identical. The reliability of component ijk is defined to be the probability that component ijk is functioning at the end of the next mission and given by

$$R_{ijk} = \Pr(Y_{ijk}^+ = 1) = r_j X_{ijk}$$

Equation 3.8

Likewise, the reliability of subsystem ij is defined to be the probability that subsystem ij is functioning at the end of the next mission and given by

$$R_{ij} = \Pr(Y_{ij}^+ = 1) = \prod_{k=1}^{n_j} R_{ijk} = 1 - \prod_{k=1}^{n_j} (1 - R_{ijk})$$

Equation 3.9

Since the components within a subsystem are identical, Equation 3.9 can be simplified. Let b_{ij} denote the number of functioning components in subsystem ij at the beginning of the next mission. Note that

$$b_{ij} = \sum_{k=1}^{n_j} X_{ijk}$$

Equation 3.10

Then

$$R_{ij} = 1 - (1 - r_j)^{b_{ij}}$$

Equation 3.11

Finally, the reliability of system i is defined to be the probability that system i is functioning at the end of the next mission and given by

$$R_i = \prod_{j=1}^m R_{ij}$$

Equation 3.12

3.3 Selective Maintenance Model for a Set of Systems

Recall that we consider the case in which a set of missions has just ended and all systems have returned to their base of operation and maintenance. The base possesses the technological capability to repair any failed component and ideally, all failed components would be repaired prior to the beginning of the next mission; however, maintenance resource limitations may prevent the repair of all failed components.

Let a_{ij} denote the number of failed components in subsystem ij , and note that a_{ij} is given by

$$a_{ij} = \sum_{k=1}^{n_j} (1 - Y_{ijk})$$

Equation 3.13

Furthermore, the total number of failed components of type j is given by

$$a_{\bullet j} = \sum_{i=1}^q a_{ij}$$

Equation 3.14

Each repair consumes a fixed amount of each of s maintenance resources (for example, labor, equipment, spare parts). Let α_{jl} denote the amount of resource l consumed by repairing a component of type j , and suppose β_l denotes the amount of resource l available during a single break. If

$$\sum_{j=1}^m \alpha_{jl} a_{\bullet j} \leq \beta_l$$

Equation 3.15

for all $l = 1, 2, \dots, s$, then all failed components may be repaired prior to the next mission. Otherwise, a method is needed to decide which failed components should be repaired and which components must remain in a failed condition. This decision-making process is the selective maintenance scenario that we consider.

Let d_{ij} denote the number of failed components in subsystem ij to be repaired prior to the beginning of the next set of missions. Then

$$d_{ij} = \sum_{k=1}^{n_j} (X_{ijk} - Y_{ijk})$$

Equation 3.16

The total number of failed components of type j repaired prior to the next set of missions is given by

$$d_{\bullet j} = \sum_{i=1}^q d_{ij}$$

Equation 3.17

To address the selective maintenance issue, we formulate a nonlinear, discrete optimization model where the d_{ij} values serve as the decision variables.

The first constraint on these decision variables is that they be integer-valued. Second, the number of repairs is limited to the number of failed components, that is,

$$0 \leq d_{ij} \leq a_{ij}$$

Equation 3.18

Third, the number of repairs is limited by the available maintenance resources, that is,

$$\sum_{j=1}^m \alpha_{jl} d_{\bullet j} \leq \beta_l$$

Equation 3.19

for all $l = 1, 2, \dots, s$. The objective in choosing values for the decision variables is to maximize the probability that all upcoming missions are completed successfully. This probability is referred to as overall reliability and given by

$$R = \prod_{i=1}^q R_i = \prod_{i=1}^q \prod_{j=1}^m 1 - (1 - r_j)^{b_{ij}}$$

Equation 3.20

where

$$b_{ij} = n_j - a_{ij} + d_{ij}$$

Equation 3.21

The resulting optimization model is given by

$$\begin{aligned}
\text{P:} \quad & \text{Maximize} \quad R = \prod_{i=1}^q \prod_{j=1}^m 1 - (1 - r_j)^{n_j - a_{ij} + d_{ij}} \\
\text{subject to} \quad & \sum_{i=1}^q \sum_{j=1}^m \alpha_{ij} d_{ij} \leq \beta_l \quad l = 1, 2, \dots, s \\
& 0 \leq d_{ij} \leq a_{ij}, \text{ integer} \quad i = 1, 2, \dots, q \\
& j = 1, 2, \dots, m
\end{aligned}$$

Equation 3.22

As a numerical example, consider the system presented in *Figure 1* having $q = 2$ and $m = 3$. Two ($s = 2$) limited maintenance resources limit repair activities during the break between mission sets. Note that $\beta_1 = 16$ and $\beta_2 = 10$. The remaining parameters for the set of systems are given in *Table 3.1*. Suppose all systems have just returned from a set of missions. The status of the set of systems is summarized in *Table 3.2* and *Figure 2*. Note that both systems successfully completed the last mission. Repairing all the failed components would require 18 units of resource 1 and 11 units of resource 2. Therefore, resource 1 is limited and we can use the selective maintenance model **P** to determine which components to repair to maximize the overall reliability. By enumerating all feasible solutions, we identify the optimal selective maintenance solution for this scenario. This solution is summarized in *Table 3.3*, and the revised status of the set of systems based on the solution is shown in *Figure 3*. If no maintenance were performed, the overall reliability would be 0.5688, and if sufficient maintenance resources were available to repair all failed components, the overall reliability would be 0.9665.

Table 3.1: Model Parameters for Example Set of Systems

Subsystem	n_j	r_j	α_{j1}	α_{j2}
1	2	0.90	2	3
2	3	0.85	3	1
3	2	0.94	2	2

Table 3.2: Failed Components for Example Set of Systems

System (q)	Subsystem (i)		
	1	2	3
1	$a_{11} = 1$	$a_{12} = 2$	$a_{13} = 1$
2	$a_{21} = 0$	$a_{22} = 2$	$a_{23} = 1$

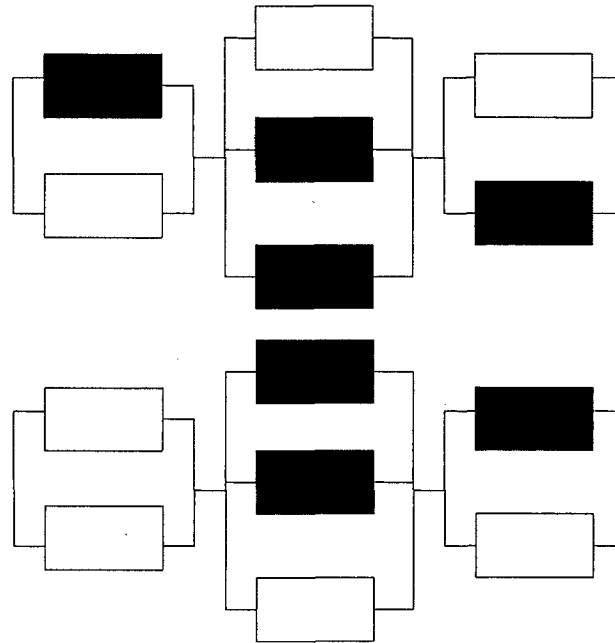


Figure 2: Status Prior to Mission

Table 3.3: Optimal Selective Maintenance Actions

System	d_{11}	d_{12}	d_{13}	Reliability	Fleet Reliability
1	1	2	1	0.9831	0.9479
2	0	1	1	0.9642	

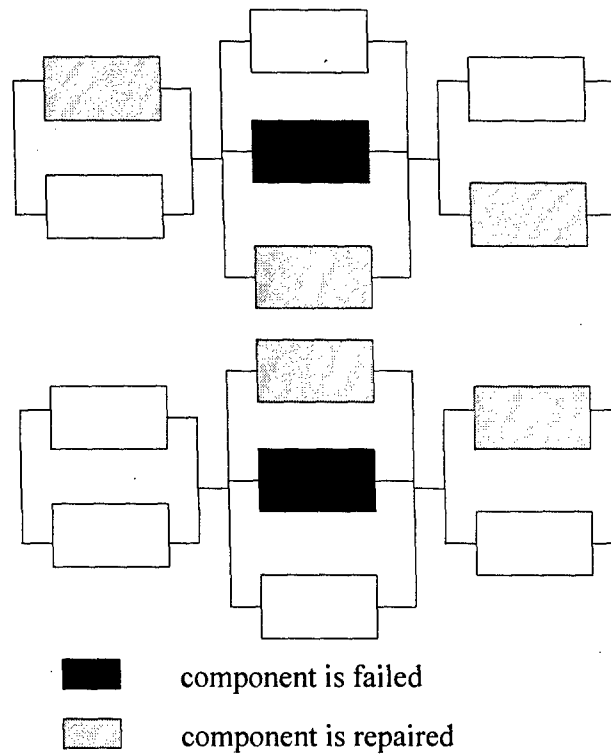


Figure 3: Status after Maintenance

3.4 Solution Procedures

For small scenarios such as the one presented, Visual Basic[®] code that enumerates all feasible solutions is executed in a macro within a Microsoft[®] Excel spreadsheet. The code generates a solution (specifically, the d_{ij} values), and the solution's feasibility is checked against the model constraints. If the solution is infeasible, it is disregarded; however, if a feasible solution is generated, the resulting overall reliability is calculated. If the overall reliability of the solution is equal to or greater than the largest reliability computed previously, that solution (and its corresponding overall reliability) is written to a file and a new solution is generated. After generating all possible solutions, the optimal solution (or all optimal solutions in the case of a tie) is output to an Excel worksheet. This enumeration code is able to handle any scenario size and guarantees the identification of all optimal solutions. Also, solution feasibility checks and computations are done "on-the-fly," eliminating the excessive use of computer memory.

Although the enumeration code guarantees an optimal solution, scenarios involving more than four systems result in lengthy computational times. Therefore, the use of a search-based heuristic based on the

concepts of GAs is applied to larger scenarios. GAs are computationally intense search methods originally applied in the area of artificial intelligence. Over the past decade, increases in computer speeds have made this directed trial-and-error method increasingly popular. GAs operate on the principle of "survival of the fittest." In general, GA keeps track of a fixed population of candidate solutions called chromosomes. Each element, or gene, within a chromosome represents some portion of the solution. The solutions are then ranked using a fitness function related to the objective function of the model. The best solutions are reserved for crossover and inferior solutions are eliminated.

Crossover of the best solutions creates chromosomes to replace the ones that were eliminated from the initial population. Two chromosomes (parents) are chosen for crossover, which results in two new chromosomes (children). Consider a crossover example involving the chromosomes shown in *Figure 4*. Let the 3rd gene denote the position of the crossover and suppose the length of the crossover is 3. As shown in *Figure 5*, the first two genes from parent one become the first two genes in child one, and the last three genes from parent two become the last three genes in child one. The opposite is also true. That is, the first two genes from parent two become the first two genes in child two, and the last three genes from parent one become the last three genes in child two. The crossover routine and resulting child chromosomes are shown in *Figure 5*.



Figure 4: Parent Chromosomes Chosen for Crossover

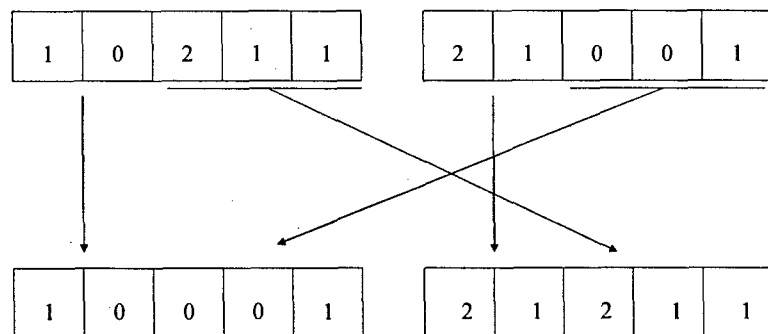


Figure 5: Crossover Routine and Resulting Child Chromosomes

Occasionally, a GA may become “stuck” on a local optimal solution. To force the algorithm to leave local optima in search of better solutions, some of the new chromosomes undergo mutation. Mutation is performed by selecting a gene within the chromosome and assigning a new value for that gene. New chromosomes created with crossover and mutation are merged with the parent chromosomes to create a new generation. The GA is executed for a specified number of generations.

In the GA used to evaluate model **P**, the individual genes within a chromosome represent the number of components to repair in subsystem ij (that is, the decision variable, d_{ij}). Therefore, each chromosome contains $q * m$ genes. An initial population of candidate solutions (chromosomes) is created. The size of the population is an input parameter specified by the model user. To initialize the population, a specified number of chromosomes are created such that the value of each gene is a discrete uniform random variable over the integers $\{0, 1, \dots, a_{ij}\}$. The population is initialized exactly once for each scenario evaluated.

The initial population represents the first generation. The feasibility of each chromosome in the generation is checked against model constraints. If a chromosome is found to be infeasible, it undergoes mutation. That is, a gene is chosen at random, and the new value of the gene is initialized. The chromosome continues to undergo mutation until it is feasible. Once a generation of feasible solutions exists, each chromosome is ranked according to the fitness function. In this case, the fitness function is the overall reliability resulting from each chromosome. The best solutions, or top-half of the ranked solutions, are reserved for crossover, and the inferior solutions are eliminated.

In the crossover routine, two parent chromosomes are chosen at random. The position and length of the crossover are also chosen randomly. The crossover is performed as shown in *Figure 4* and *Figure 5*. For each chromosome created in the crossover routine, the probability of mutation is defined to be 0.1. That is, each chromosome has a 10% chance of undergoing mutation. Once crossover and mutation are complete, the parents and children are merged to create a new generation.

The new generation is ranked according to overall reliability. The best solutions are subjected to crossover, and chromosomes resulting from crossover are subjected to mutation. The parents and children are then merged, forming a new generation. This process is repeated for a user-specified number of generations. This logic is summarized in *Figure 6*. This approach uses an application developed in Visual Basic® and programmed as a macro within an Excel spreadsheet. Instructions on using both the total enumeration strategy and the GA for solving model **P** are presented in Appendix A.

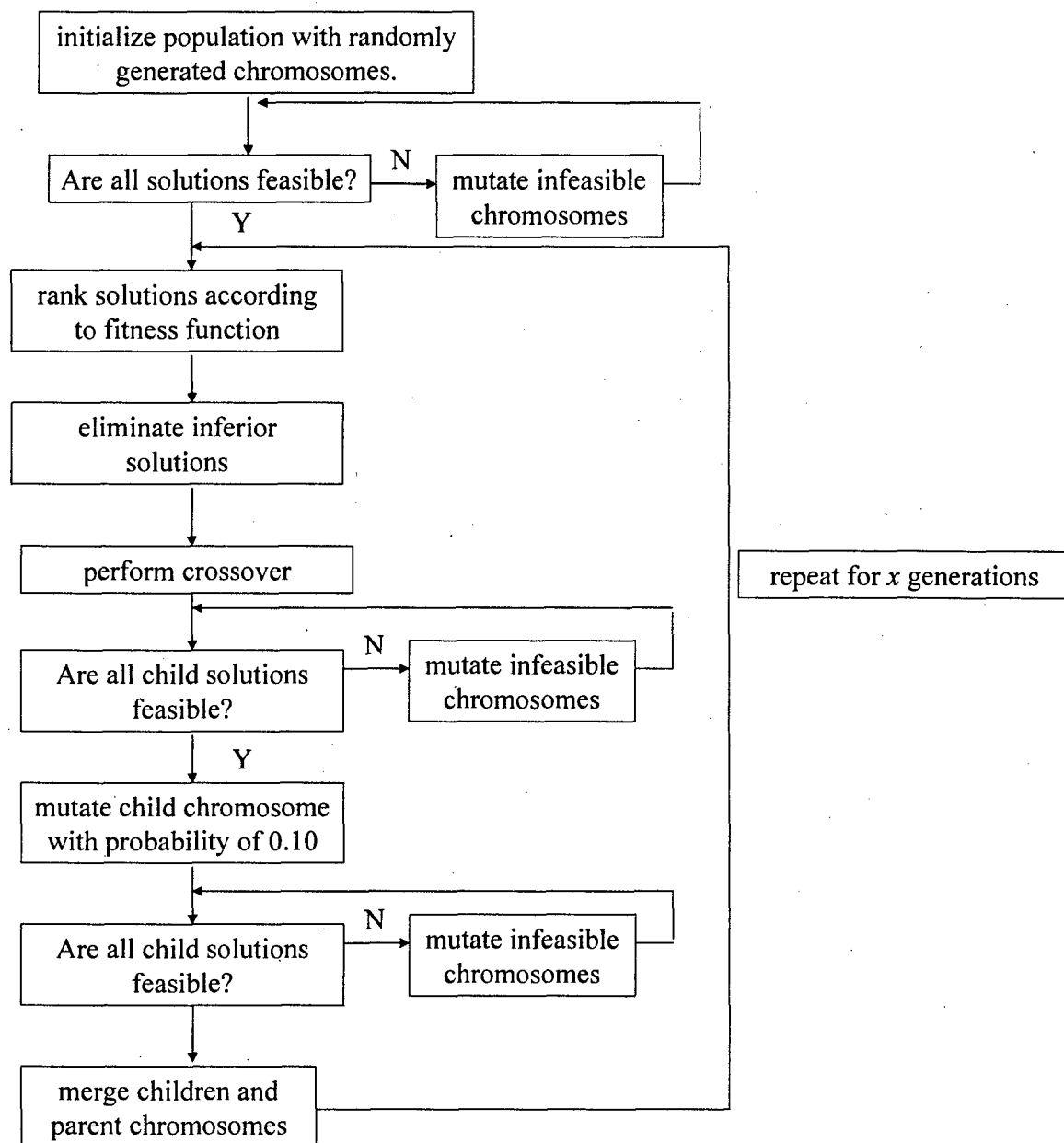


Figure 6: GA Logic

4 The Static Model

The scenario captured by the optimization model **P** addresses selective maintenance for a set of systems, but it does not capture the assignment of systems to missions. In the scenario address by model **P**, assignment is not relevant, because all missions in the upcoming set are identical. In this section, we add assignment to the model by considering the scenario in which the missions in the upcoming set are not identical.

4.1 Hypothetical Set of Systems

Consider a set of q independent and identical systems. Each system is comprised of m independent subsystems connected in series, and each subsystem, say subsystem j , contains n_j independent and identical copies of a CFR component connected in parallel. A graphical representation (reliability block diagram) of an example set of systems is shown in *Figure 7*. Note that each component in *Figure 7* is labeled " i, j, k ," where i denotes the system number, j denotes the subsystem number and k denotes the component number.

The set of systems is required to perform sequential sets of missions that originate from and return to a common base of operation. Each set of missions contains q independent missions with common start times; however, the missions have different durations. At any time, a specific component, subsystem or system is in one of two states: functioning (1) or failed (0). Components only fail during missions and failed components can be repaired only during the breaks between sets of missions. Note that all maintenance is performed at the base.

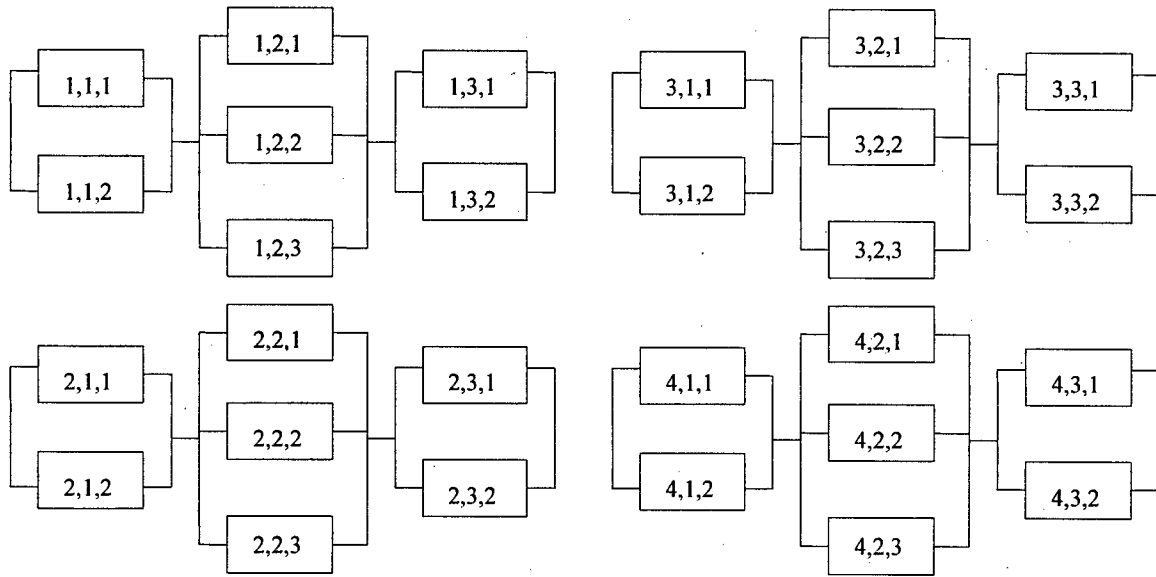


Figure 7: Example Set of Systems

4.2 System Performance

Suppose that a set of missions has just ended and all systems have returned to their base of operation and maintenance. Note that component ijk refers to component k in subsystem j of system i , and note that subsystem ij refers to subsystem j in system i . Let Y_{ijk} denote the status of component ijk , let Y_{ij} denote the status of subsystem ij , and let Y_i denote the status of system i . Since each subsystem is a parallel arrangement of its components,

$$Y_{ij} = \prod_{k=1}^{n_j} Y_{ijk} = 1 - \prod_{k=1}^{n_j} (1 - Y_{ijk})$$

Equation 4.1

Since each system is a series arrangement of its subsystems,

$$Y_i = \prod_{j=1}^m Y_{ij}$$

Equation 4.2

Note that if $Y_{ij} = 0$ for some $j = 1, 2, \dots, m$, then $Y_i = 0$ and system i did not successfully complete the last mission.

Prior to the next set of missions, maintenance (repair) can be performed on some or all of the failed components. Let X_{ijk} denote the status of component ijk at the start of the next mission, let X_{ij} denote the status of subsystem ij at the start of the next mission, and let X_i denote the status of system i at the start of the next mission. Note that

$$X_{ij} = \prod_{k=1}^{n_j} X_{ijk}$$

Equation 4.3

and

$$X_i = \prod_{j=1}^m X_{ij}$$

Equation 4.4

Furthermore, note that if $Y_{ijk} = 0$ and $X_{ijk} = 1$, then component ijk was repaired during the break between missions. Also, note that if $X_{ij} = 0$ for some $j = 1, 2, \dots, m$, then $X_i = 0$ and system i is unable to undertake its next mission.

Clearly, the X_{ijk} values have a direct impact on the status of components, subsystems and systems at the end of the upcoming mission. Let Y_{ijk}^+ denote the status of component ijk at the end of the next mission, let Y_{ij}^+ denote the status of subsystem ij at the end of the next mission, and let Y_i^+ denote the status of system i at the end of the next mission. Note that

$$Y_{ij}^+ = \prod_{k=1}^{n_j} Y_{ijk}^+$$

Equation 4.5

and

$$Y_i^+ = \prod_{j=1}^m Y_{ij}^+$$

Equation 4.6

Also, note that if $X_{ijk} = 0$, then $Y_{ijk}^+ = 0$. If $X_{ijk} = 1$, then Y_{ijk}^+ is a random variable. Assuming $X_i = 1$, then both Y_{ij}^+ for all $j = 1, 2, \dots, m$ and Y_i^+ , $i = 1, 2, \dots, q$ are random variables. Furthermore, note that if $Y_{ij}^+ = 0$ for some $j = 1, 2, \dots, m$, then $Y_i^+ = 0$ and system i did not successfully complete the next mission.

Since the upcoming missions are not identical, the length of the mission must be incorporated into the component reliability calculation. In other words, we must adjust the component reliability values depending upon the mission assignment. Let λ_j denote the failure rate of a type j component, and let ρ_{ij} denote the probability that a functioning type j component completes its next mission if it is assigned mission i . Let t_i denote the length of mission i , $i = 1, 2, \dots, q$. Then

$$\rho_{ij} = e^{-\lambda_j t_i}$$

Equation 4.7

This parameter quantifies the difficulty of each mission and the resulting differences between missions. Here, we consider mission length to be synonymous with mission difficulty, but other measures of difficulty could be applied by adjusting the values of ρ_{ij} .

Next, we define conditional component reliability, r_{ij} as the probability that a functioning component in subsystem ij completes its next mission

$$r_{ij} = \sum_{i'=1}^q \rho_{i'j} g_{ii'}$$

Equation 4.8

where $g_{ii'}$ denotes the mission assignment variable and is given by

$$g_{ii'} = \begin{cases} 1 & \text{if system } i \text{ is assigned to mission } i' \\ 0 & \text{otherwise} \end{cases}$$

Equation 4.9

Note that conditional component reliability is only indexed on i and j because the components within a subsystem are identical. The resulting measures of component and subsystem reliability are given by

$$R_{ijk} = r_{ij} X_{ijk}$$

Equation 4.10

$$R_{ij} = 1 - (1 - r_{ij})^{b_{ij}}$$

Equation 4.11

where

$$b_{ij} = n_j - a_{ij} + d_{ij}$$

Equation 4.12

Finally, the reliability of system i is defined to be the probability that system i is functioning at the end of the next mission and given by

$$R_i = \prod_{j=1}^m R_{ij}$$

Equation 4.13

4.3 Formulation of the Static Model

Recall that we consider the case in which a set of missions has just ended and all systems have returned to their base of operation and maintenance. The base possesses the technological capability to repair any failed component and ideally, all failed components would be repaired prior to the beginning of the next mission; however, maintenance resource limitations may prevent the repair of all failed components.

Let a_{ij} denote the number of failed components in subsystem ij , and note that a_{ij} is given by

$$a_{ij} = \sum_{k=1}^{n_j} (1 - Y_{ijk})$$

Equation 4.14

Furthermore, the total number of failed components of type j is given by

$$a_{\bullet j} = \sum_{i=1}^q a_{ij}$$

Equation 4.15

Each repair consumes a fixed amount of each of s maintenance resources (labor, equipment, and spare parts, for example.). Let α_{jl} denote the amount of resource l consumed by repairing a component of type j , and suppose β_l denotes the amount of resource l available during a single break. If

$$\sum_{j=1}^m \alpha_{jl} a_{\bullet j} \leq \beta_l$$

Equation 4.16

for all $l = 1, 2, \dots, s$, then all failed components may be repaired prior to the next mission. Otherwise, a method is needed to decide which failed components should be repaired and which components must remain in a failed condition. This is the selective maintenance scenario that we consider.

Let d_{ij} denote the number of failed components in subsystem ij to repair prior to the beginning of the next mission. Then

$$d_{ij} = \sum_{k=1}^{n_j} (X_{ijk} - Y_{ijk})$$

Equation 4.17

The total number of failed components of type j repaired prior to the next mission is given by

$$d_{\bullet j} = \sum_{i=1}^q d_{ij}$$

Equation 4.18

To incorporate assignment into the model, the mission assignment variable, g_{ir} , is defined. To address the selective maintenance issue, we formulate a nonlinear, discrete optimization model where the d_{ij} and g_{ir} values are the decision variables.

The first constraint on the d_{ij} decision variables is that they be integer-valued. Second, the number of repairs is limited to the number of failed components, that is,

$$0 \leq d_{ij} \leq a_{ij}$$

Equation 4.19

Third, the number of repairs is limited by the available maintenance resources, specifically,

$$\sum_{j=1}^m \alpha_{jl} d_{\bullet j} \leq \beta_l$$

Equation 4.20

for all $l = 1, 2, \dots, s$.

The first constraint on the $g_{ii'}$ decision variables is that they are binary variables. Second, a system must be assigned exactly one mission, that is,

$$\sum_{i'=1}^q g_{ii'} = 1$$

Equation 4.21

Third, each mission must be assigned to exactly one system, specifically,

$$\sum_{i=1}^q g_{ii'}$$

Equation 4.22

The objective in choosing the values for the decision variables is to maximize the probability that all upcoming missions are completed successfully. This probability is referred to as overall reliability and is given by

$$R = \prod_{i=1}^q R_i = \prod_{i=1}^q \prod_{j=1}^m 1 - (1 - r_{ij})^{h_{ij}}$$

Equation 4.23

The full formulation of the optimization model for this scenario is as follows:

$$\text{SP: Maximize } R = \prod_{i=1}^q \prod_{j=1}^m 1 - \left(1 - \sum_{i'=1}^q e^{-\lambda_j t_{i'}} g_{ii'} \right)^{n_j - a_{ij} + d_{ij}}$$

$$\text{subject to } \sum_{i=1}^q \sum_{j=1}^m \alpha_{ji} d_{ij} \leq \beta_l \quad l = 1, 2, \dots, s$$

$$\sum_{i=1}^q g_{ii'} = 1$$

$$\sum_{i'=1}^q g_{ii'} = 1$$

$$g_{ii'}, \text{ binary} \quad i = 1, 2, \dots, q$$

$$i' = 1, 2, \dots, q$$

$$0 \leq d_{ij} \leq a_{ij}, \text{ integer} \quad i = 1, 2, \dots, q$$

$$j = 1, 2, \dots, m$$

Equation 4.24

4.4 Solution Procedure

Due to the complexity of incorporating both selective maintenance and mission assignment into the model, we prefer to use a search-based heuristic such as GAs rather than total enumeration. GAs are computationally intense search methods originally applied in the area of artificial intelligence. Over the past decade, increases in computer speeds have made this directed trial-and-error method increasingly popular. GAs operate on the principle of “survival of the fittest.” In general, GA keeps track of a fixed population of candidate solutions called chromosomes. Each element, or gene, within a chromosome represents some portion of the solution. The solutions are then ranked using a fitness function related to the objective function of the model. The best solutions are reserved for crossover, and inferior solutions are eliminated.

Crossover of the best solutions creates chromosomes to replace the ones that were eliminated from the initial population. Two chromosomes (parents) are chosen for crossover, which results in two new chromosomes (children). This combination of the parents and children is referred to as a generation.

Occasionally, a GA may become “stuck” on a local optimal solution. To force the algorithm to leave local optima in search of better solutions, some of the new chromosomes undergo mutation. Mutation is performed by selecting a gene within the chromosome and assigning a new value for that gene. New chromosomes created with crossover and mutation are merged with the parent chromosomes to create a new generation. The GA repeats itself for a specified number of generations.

In the GA used to evaluate model SP, the individual genes within a chromosome represent not only the mission assignment (that is, the decision variable g_{ij}) but also the number of components to repair in subsystem ij (specifically, the decision variable d_{ij}). Therefore, each chromosome contains $q + q * m$ genes. An initial population of candidate solutions (chromosomes) is created. The size of the population is an input parameter specified by the model user. To initialize the population, a specified number of chromosomes are created such that the value of the first q genes are discrete uniform random variables over the integers $\{1, 2, \dots, q\}$. The values of these genes are generated by randomly selecting one of the $q!$ feasible mission assignments. The value of the remaining $q*m$ genes represent the selective maintenance decisions. The value of these genes are generated as discrete uniform random variables over the integers $\{0, 1, \dots, a_{ij}\}$. The population is initialized exactly once for each scenario evaluated.

The initial population represents the first generation. The feasibility of each chromosome in the generation is checked against model constraints. If a chromosome is found to be infeasible, two checks are made. First, the model checks to see if any duplicate mission assignments occur in the first q genes. If a duplicate mission assignment is found, one of the duplicate genes is chosen at random and initialized again. The chromosome is checked again for duplicate missions, and the above process is repeated until no duplicate mission assignments exist. Second, the values of the remaining $q*m$ chromosomes may violate model constraints. If this is the case, the chromosome continues to undergo mutation until it is feasible. That is, one gene is chosen at random and initialized again. Once a generation of feasible solutions exists, each chromosome is ranked according to a fitness function. In this case, the fitness function is the overall reliability resulting from each chromosome. The best solutions are reserved for crossover, and the inferior solutions are eliminated.

In the crossover routine, two parent chromosomes are chosen at random. In this scenario, crossover occurs in two steps. First, crossover is performed on the first q genes representing the mission assignment. After this portion of crossover is complete, the genes are checked for duplicate assignments. Considering the chromosomes shown in *Figure 8*, suppose the position of the crossover is 1, and the length of the crossover is 2. The chromosomes after the first crossover step are shown in *Figure 9*.

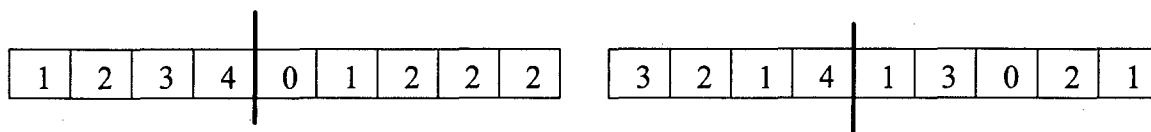


Figure 8: Parent Chromosomes Chosen for Crossover

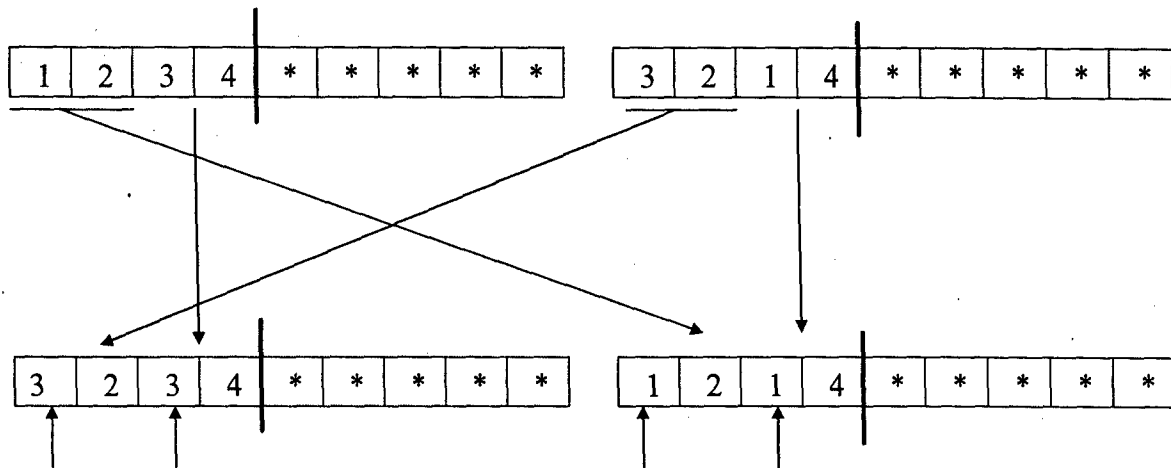


Figure 9: Step One of Crossover Routine

Notice that in each of the new chromosomes, a mission assignment is duplicated. Specifically, the chromosome on the right has two systems assigned to mission three and none to mission one, and the chromosome on the left has two systems assigned to mission one and none to mission three. To correct this situation, a duplicate assignment gene is chosen from each chromosome. The value of that gene is initialized again. The chromosome is checked again for duplicate missions, and the above process is repeated until no duplicate mission assignments exist.

Next, the remaining $q*m$ genes undergo crossover. The position and length of the crossover are also chosen randomly. Again, considering the chromosomes shown in *Figure 8*, suppose the position of the crossover is gene 2 (in the second section of the chromosome) and the length of the crossover is 3. *Figure 10* shows the second step in the crossover routine.

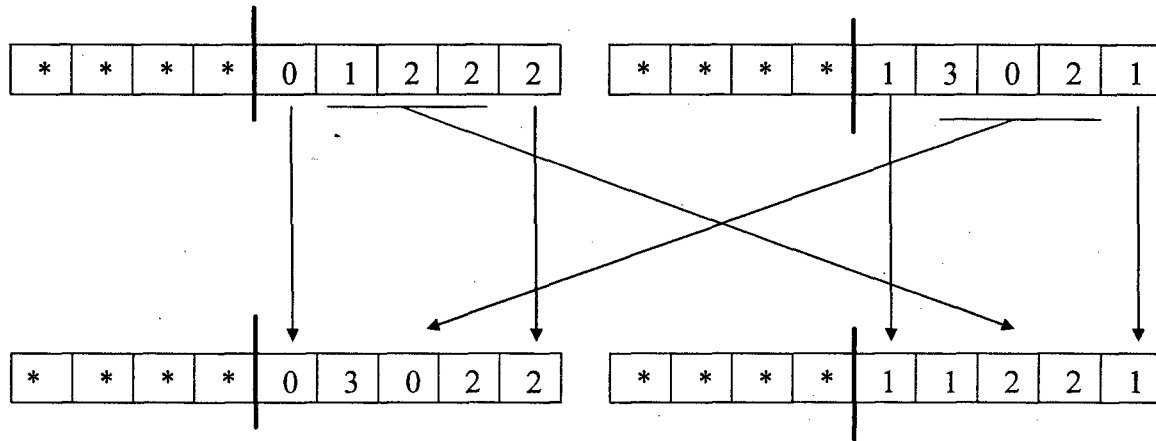


Figure 10: Step Two of Crossover Routine

For each chromosome created in the crossover routine, the probability of mutation is defined to be 0.1. That is, each chromosome has a 10% chance of undergoing mutation. Once crossover and mutation are complete, the parents and children are merged to create a new generation.

The new generation is ranked according to overall reliability. The best solutions are subjected to crossover, and chromosomes resulting from crossover are subjected to mutation. The parents and children are then merged, forming a new generation. This process is repeated for a user-specified number of generations.

As a numerical example, consider the system presented in *Figure 11*, having $q = 4$ and $m = 3$. Two ($s = 2$) limited maintenance resources limit repair activities. Note that $\beta_1 = 16$ and $\beta_2 = 14$. The next mission durations (in hours) are $t_1 = 4$, $t_2 = 1.75$, $t_3 = 2$, $t_4 = 0.5$. The remaining parameters for the set of systems are given in *Table 4.1*. To repair all failed components requires 30 units of resource 1 and 21 units of resource 2. Therefore, we can use our selective maintenance model to determine the mission assignment for each system and which components to repair to maximize the overall reliability. We solve this scenario using the GA. This approach uses an application developed in Visual Basic[®] and programmed as a macro within a Microsoft[®] Excel spreadsheet. Instructions for using the spreadsheet appear in *Appendix B*. *Figure 12* shows the system status after maintenance. The overall reliability of the set of systems with the proposed repairs and mission assignment is 0.5347. Given this mission assignment, overall reliability would be 0.6039 if sufficient maintenance resources were available to repair all failed components.

Table 4.1: Model Parameters for Example Set of Systems

Subsystem	n_j	λ_j (in failures per hour)	α_{j1}	α_{j2}	a_{1j}	a_{2j}	a_{3j}	a_{4j}
1	2	0.10	2	3	1	1	1	0
2	3	0.16	3	1	1	2	2	1
3	2	0.06	2	2	1	0	1	1

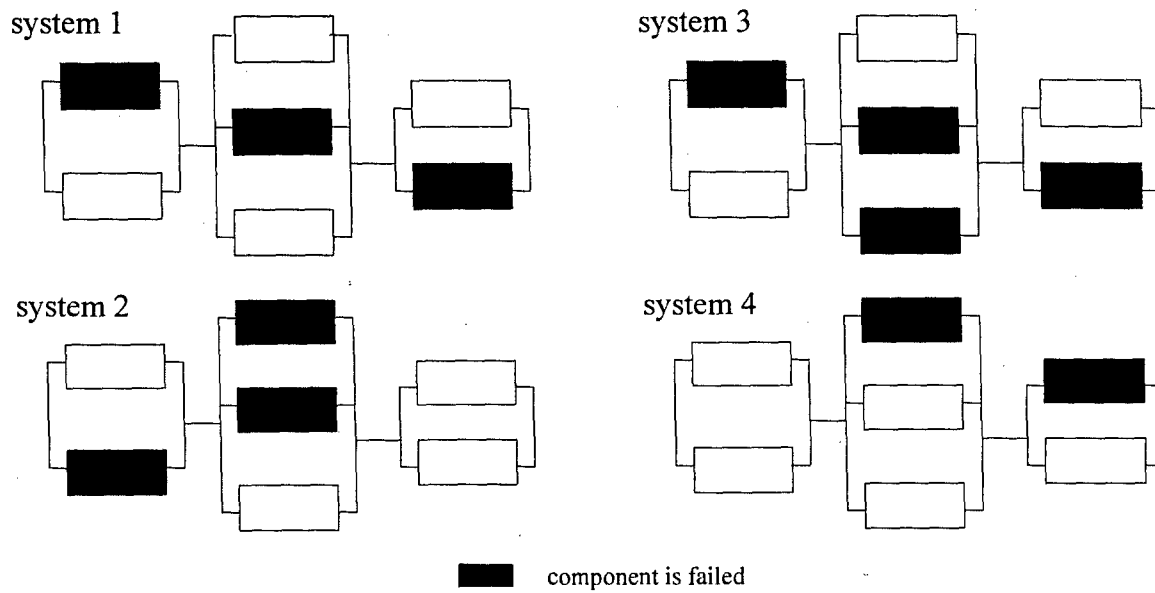


Figure 11: Status Prior to Maintenance

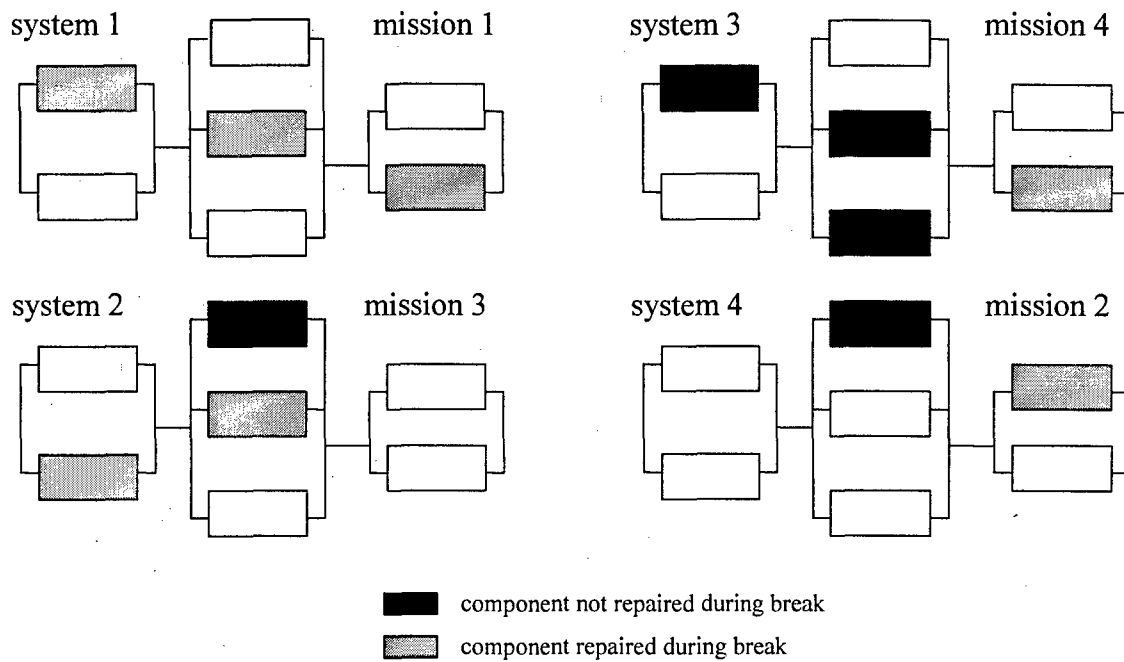


Figure 12: Mission Assignment and Status after Maintenance

5 The Dynamic Model

The scenario captured by the optimization model **P** addresses selective maintenance for a set of systems but not assignment. Assignment is not addressed because all missions in the upcoming set are identical. In *Section 4*, we add assignment to the model by considering the scenario in which the missions in the upcoming set are not identical; however, all missions within a set have common start times. In this scenario, missions start and end at different times, and maintenance and scheduling decisions are made over a series of time “buckets.”

5.1 Hypothetical Set of Systems

Consider a set of q independent and identical systems. Each system is comprised of m independent subsystems connected in series, and each subsystem, say subsystem j , contains n_j independent and identical copies of a CFR component connected in parallel. A graphical representation (reliability block diagram) of an example set of systems is shown in *Figure 13*. Note that each component in *Figure 13* is labeled “ i, j, k ,” where i denotes the system number, j denotes the subsystem number and k denotes the component number.

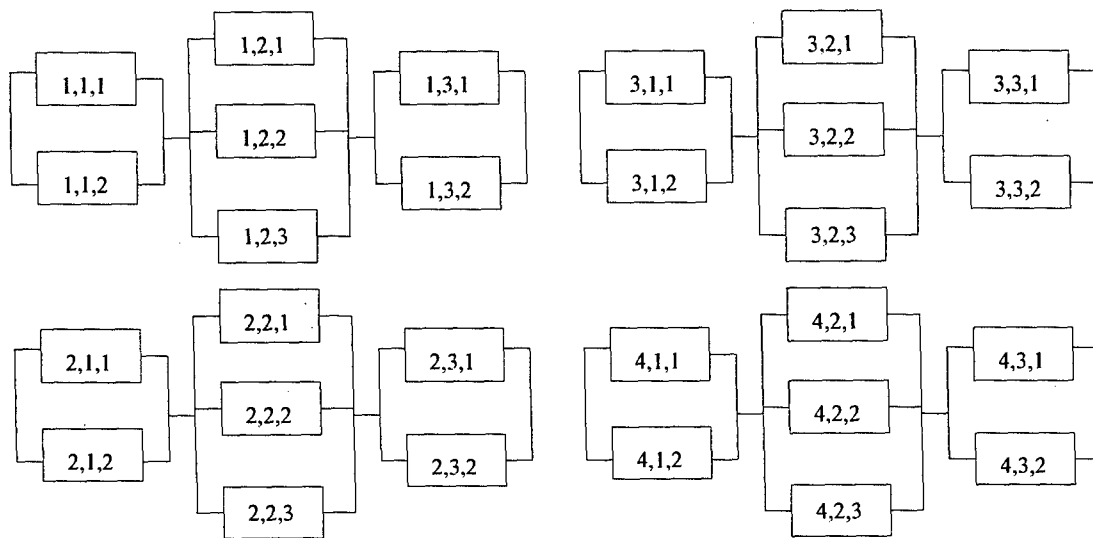


Figure 13: Example Set of Systems

The set of systems is required to perform sequential sets of missions that originate from and return to a common base of operation. Each set of missions contains q independent missions. Missions have varying

start times and durations. Additionally, systems do not necessarily complete their previous missions at the same time. At any time, a specific component, subsystem or system is in one of two states: functioning (1) or failed (0). Components only fail during missions, and failed components can be repaired only during the breaks between sets of missions. Note that all maintenance is performed at the base.

5.2 System Performance

Suppose that system i returns to its base of operation and maintenance from its current mission at the beginning of time bucket RT_i . We define "now" as the beginning of time bucket 1, and at least one system has $RT_i \leq 1$. During the planning horizon, all systems not initially at the base will return from their current mission.

Note that component ijk refers to component k in subsystem j of system i , and note that subsystem ij refers to subsystem j in system i . Let Y_{ijk} denote the status of component ijk , let Y_{ij} denote the status of subsystem ij , and let Y_i denote the status of system i . Since each subsystem is a parallel arrangement of its components,

$$Y_{ij} = \prod_{k=1}^{n_j} Y_{ijk} = 1 - \prod_{k=1}^{n_j} (1 - Y_{ijk})$$

Equation 5.1

Since each system is a series arrangement of its subsystems,

$$Y_i = \prod_{j=1}^m Y_{ij}$$

Equation 5.2

Note that if $Y_{ij} = 0$ for some $j = 1, 2, \dots, m$, then $Y_i = 0$ and system i did not successfully complete the last mission.

Prior to the system's next assigned mission, maintenance (repair) can be performed on some or all of the failed components. Let X_{ijk} denote the status of component ijk at the start of the next mission, let X_{ij} denote the status of subsystem ij at the start of the next mission, and let X_i denote the status of system i at the start of the next mission. Note that

$$X_{ij} = \prod_{k=1}^{n_j} X_{ijk}$$

Equation 5.3

and

$$X_i = \prod_{j=1}^m X_{ij}$$

Equation 5.4

Furthermore, note that if $Y_{ijk} = 0$ and $X_{ijk} = 1$, then component ijk was repaired during the break between missions. Also, note that if $X_{ij} = 0$ for some $j = 1, 2, \dots, m$, then $X_i = 0$ and system i is unable to undertake its next mission.

Clearly, the X_{ijk} values have a direct effect on the status of components, subsystems and systems at the end of the upcoming mission. Let Y_{ijk}^+ denote the status of component ijk at the end of the next mission, let Y_{ij}^+ denote the status of subsystem ij at the end of the next mission, and let Y_i^+ denote the status of system i at the end of the next mission. Note that

$$Y_{ij}^+ = \prod_{k=1}^{n_j} Y_{ijk}^+$$

Equation 5.5

and

$$Y_i^+ = \prod_{j=1}^m Y_{ij}^+$$

Equation 5.6

Also, note that if $X_{ijk} = 0$, then $Y_{ijk}^+ = 0$. If $X_{ijk} = 1$, then Y_{ijk}^+ is a random variable. Assuming $X_i = 1$, then both Y_{ij}^+ for all $j = 1, 2, \dots, m$ and Y_i^+ , $i = 1, 2, \dots, q$ are random variables. Furthermore, note that if

$Y_{ij}^+ = 0$ for some $j = 1, 2, \dots, m$, then $Y_i^+ = 0$ and system i did not successfully complete the next mission.

Each system within the set must be assigned exactly one upcoming mission. Mission i' starts at the beginning of time bucket $ST_{i'}$, and $t_{i'}$ denotes the length of mission i' . Note that no future mission ends before the end of the planning horizon. Since the upcoming missions are not identical, the length of the mission must be incorporated into the component reliability calculation. In other words, we must adjust the component reliability values depending upon the mission assignment. Let λ_j denote the failure rate of a type j component, $j = 1, 2, \dots, m$, and let $\rho_{i'j}$ denote the probability that a functioning type j component completes its next mission if it is assigned mission i' , $i' = 1, 2, \dots, q$, $j = 1, 2, \dots, m$. Let $t_{i'}$ denote the length of mission i' , $i' = 1, 2, \dots, q$.

$$\rho_{i'j} = e^{-\lambda_j t_{i'}}$$

Equation 5.7

This parameter quantifies the difficulty of each mission and the resulting differences between missions. Here, we consider mission length to be synonymous with mission difficulty, but other measures of difficulty could be applied by adjusting the values of $\rho_{i'j}$, $i' = 1, 2, \dots, q$, $j = 1, 2, \dots, m$.

Next, we define conditional component reliability, r_{ij} , as the probability that a functioning component in subsystem ij completes its next mission, $i = 1, 2, \dots, q$, $j = 1, 2, \dots, m$

$$r_{ij} = \sum_{i'=1}^q \rho_{i'j} g_{ii'}$$

Equation 5.8

where $g_{ii'}$ denotes the mission assignment variable and is given by

$$g_{ii'} = \begin{cases} 1 & \text{if system } i \text{ is assigned to mission } i' \\ 0 & \text{otherwise} \end{cases}$$

Equation 5.9

$i = 1, 2, \dots, q, i' = 1, 2, \dots, q$. Note that conditional component reliability is only indexed on i and j because the components within a subsystem are identical. The resulting measures of component and subsystem reliability are given by

$$R_{ijk} = r_{ij} X_{ijk}$$

Equation 5.10

and

$$R_{ij} = 1 - (1 - r_{ij})^{b_{ij}}$$

Equation 5.11

where

$$b_{ij} = n_j - a_{ij} + d_{ij}$$

Equation 5.12

Finally, the reliability of system i is defined to be the probability that system i is functioning at the end of the next mission and given by

$$R_i = \prod_{j=1}^m R_{ij}$$

Equation 5.13

5.3 Formulation of the Dynamic Model

Recall that we consider the case in which system i has returned from its current mission at the beginning of time bucket RT_i . Its base of operation and maintenance possesses the technological capability to repair any failed component and ideally, all failed components would be repaired prior to the beginning of the next mission; however, maintenance resource limitations may prevent the repair of all failed components.

Let a_{ij} denote the number of failed components in subsystem ij , and note that a_{ij} is given by

$$a_{ij} = \sum_{k=1}^{n_j} (1 - Y_{ijk})$$

Equation 5.14

Furthermore, the total number of failed components of type j is given by

$$a_{\bullet j} = \sum_{i=1}^q a_{ij}$$

Equation 5.15

Note that for systems having $RT_i > 1$, the values for the number of failed components in subsystem ij , a_{ij} are estimates. Each repair consumes a fixed amount of each of s maintenance resources (labor, equipment and spare parts, for example). Let α_{jl} denote the amount of resource l consumed by repairing a component of type j , and suppose β_l denotes the amount of resource l available during in each time bucket h . If sufficient resources exist, then all failed components may be repaired prior to the next mission. Otherwise, a method is needed to decide which failed components should be repaired and which components must remain in a failed condition. This is the selective maintenance scenario that we consider.

Let d_{ijh} denote the number of failed components in subsystem ij to repair in time bucket h . The total number (across all systems) of type j components to repair during time bucket h is given by

$$d_{\bullet jh} = \sum_{i=1}^q d_{ijh}$$

Equation 5.16

The total number of components to repair in subsystem ij during the planning horizon is given by

$$d_{ij\bullet} = \sum_{h=1}^{ST^{\max}-1} d_{ijh}$$

Equation 5.17

where

$$ST^{\max} = \max_{1 \leq i' \leq q} ST_{i'}$$

Equation 5.18

To incorporate assignment into the model, an assignment decision variable, $g_{ii'}$, is defined. To address the selective maintenance issue, we formulate a nonlinear discrete optimization model where the d_{ij} and $g_{ii'}$ values are the decision variables.

The first constraint on the d_{ij} decision variables is that they be integer-valued. Second, the number of repairs in each time bucket must be greater than or equal to zero. Third, the number of repairs is limited by the available maintenance resources, such that

$$\sum_{j=1}^m \alpha_{ji} d_{ij} \leq \beta_i$$

Equation 5.19

for all $i = 1, 2, \dots, s$.

The first constraint on the $g_{ii'}$ decision variables is that they be binary. Second, a system must be assigned exactly one mission, specifically,

$$\sum_{i=1}^q g_{ii'} = 1$$

Equation 5.20

Third, each mission must be assigned to exactly one system, that is,

$$\sum_{i'=1}^q g_{ii'}$$

Equation 5.21

Additional model constraints are necessary to restrict the time buckets in which maintenance actions and mission assignments may be performed on a particular system. First, maintenance cannot be performed if a system has not returned from its current mission. That is,

$$\sum_{i=1}^q \sum_{j=1}^m \sum_{h=1}^{RT_i-1} d_{ijh} = 0$$

Equation 5.22

for all $h = 1, 2, \dots, RT_i - 1$. Second, a system cannot be assigned a mission if it has not returned from its previous mission. This is given by

$$g_{ii'}(RT_i - ST_{i'} - 1) \leq 0$$

Equation 5.23

for all $i = 1, 2, \dots, q$ and $i' = 1, 2, \dots, q$. Finally, maintenance cannot be performed if a system has started its next mission, specifically,

$$g_{ii'}(h - ST_{i'})d_{ijh} \leq 0$$

Equation 5.24

for all $i = 1, 2, \dots, q$, $i' = 1, 2, \dots, q$, and $h = 1, 2, \dots, ST^{\max}$.

The objective in choosing values for the decision variables is to maximize the probability that all upcoming missions are completed successfully. This probability is referred to as overall reliability and is given by

$$R = \prod_{i=1}^q R_i = \prod_{i=1}^q \prod_{j=1}^m 1 - (1 - r_{ij})^{n_j - a_{ij} - \sum_h d_{ijh}}$$

Equation 5.25

The full formulation of the optimization model for this scenario is as follows:

$$\text{DSP: Maximize} \quad R = \prod_{i=1}^q \prod_{j=1}^m 1 - \left(1 - \sum_{i'=1}^q e^{-\lambda_{ji'} } \right)^{n_j - a_{ij} + \sum_h d_{ijh}}$$

$$\text{subject to} \quad \sum_{j=1}^m \alpha_{jl} d_{\bullet j} \leq \beta_l \quad l = 1, 2, \dots, s$$

$$\sum_{i=1}^q g_{ii'} = 1$$

$$\sum_{i'} g_{ii'} = 1$$

$$\sum_{i=1}^q \sum_{j=1}^m \sum_{h=1}^{RT_i-1} d_{ijh} = 0 \quad h = 1, 2, \dots, RT_i-1$$

$$g_{ii'} (RT_i - ST_{i'} - 1) \leq 0 \quad i = 1, 2, \dots, q$$

$$i' = 1, 2, \dots, q$$

$$g_{ii'} * (h - ST_{i'}) * d_{ijh} \leq 0 \quad i = 1, 2, \dots, q$$

$$i' = 1, 2, \dots, q$$

$$d_{ijh} \geq 0, \text{ integer} \quad i = 1, 2, \dots, q$$

$$j = 1, 2, \dots, m$$

$$h = 1, 2, \dots, ST^{\max}$$

$g_{i'}$, binary

$i = 1, 2, \dots, q$

$i' = 1, 2, \dots, q$

5.4 Solution Procedure

Due to the complexity of incorporating both selective maintenance and a more dynamic mission profile into the model, we prefer to use a search-based heuristic such as GAs rather than total enumeration. GAs are computationally intense search methods originally applied in the area of artificial intelligence. Over the past decade, increases in computer speeds have made this directed trial-and-error method increasingly popular. GAs operate on the principle of “survival of the fittest.” In general, GA keeps track of a fixed population of candidate solutions called chromosomes. Each element, or gene, within a chromosome represents some portion of the solution. The solutions are then ranked using a fitness function related to the objective function of the model. The best solutions are reserved for crossover and inferior solutions are eliminated.

Crossover of the best solutions creates chromosomes to replace the ones that were eliminated from the initial population. Two chromosomes (parents) are chosen for crossover, which results in two new chromosomes (children). This combination of the parents and children is referred to as a generation.

Occasionally, a GA may become “stuck” on a local optimal solution. To force the algorithm to leave local optima in search of better solutions, some of the new chromosomes undergo mutation. Mutation is performed by selecting a gene within the chromosome and assigning a new value for that gene. New chromosomes created with crossover and mutation are merged with the parent chromosomes to create a new generation. The GA repeats itself for a specified number of generations.

In the GA used to evaluate model **DSP**, the individual genes within a chromosome represent not only the mission assignment (explicitly, the decision variable $g_{i'}$), but also the number of components to repair in subsystem ij during each time “bucket” (i.e., the decision variable d_{ijh}). Therefore, each chromosome contains $q + q * m * h$ genes. An initial population of candidate solutions (chromosomes) is created. The size of the population is an input parameter specified by the model user. To initialize the population, a specified number of chromosomes are created such that the value of the first q genes are discrete uniform random variables over the integers $\{1, 2, \dots, q\}$. The values of these genes are generated by randomly selecting one of the $q!$ feasible mission assignments. The value of the remaining $q * m * h$ genes represent the selective maintenance decisions. The value of these genes are generated as discrete uniform random

variables over the integers $\{0, 1, \dots, a_{ij}\}$. The population is initialized exactly once for each scenario evaluated.

The initial population represents the first generation. The feasibility of each chromosome in the generation is checked against model constraints. If a chromosome is found to be infeasible, two checks are made. First, the model checks to see if any duplicate mission assignments occur in the first q genes. If a duplicate mission assignment is found, one of the duplicate genes is chosen at random and initialized again. The chromosome is checked again for duplicate missions, and the above process is repeated until no duplicate mission assignments exist. Second, the values of the remaining $q*m*h$ chromosomes may violate model constraints. If this is the case, the chromosome continues to undergo mutation until it is feasible. In this case, we define procedures for mutation that attempt to find a feasible solution. That is, one gene whose value is greater than or equal to 1 is chosen at random and the value is decremented by one. Once a generation of feasible solutions exists, each chromosome is ranked according to a fitness function. In this case, the fitness function is the overall reliability resulting from each chromosome. The best solutions are reserved for crossover, and the inferior solutions are eliminated.

In the crossover routine, two parent chromosomes are chosen at random. In this scenario, crossover occurs in two steps. First, crossover is performed on the first q genes representing the mission assignment. After this portion of crossover is complete, the genes are checked for duplicate assignments. Considering the chromosomes shown in *Figure 14*, suppose the position of the crossover is 1, and the length of the crossover is 2. The chromosomes after the first crossover step are shown in *Figure 15*.

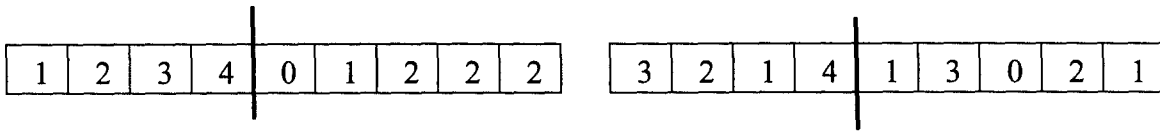


Figure 14: Parent Chromosomes Chosen for Crossover

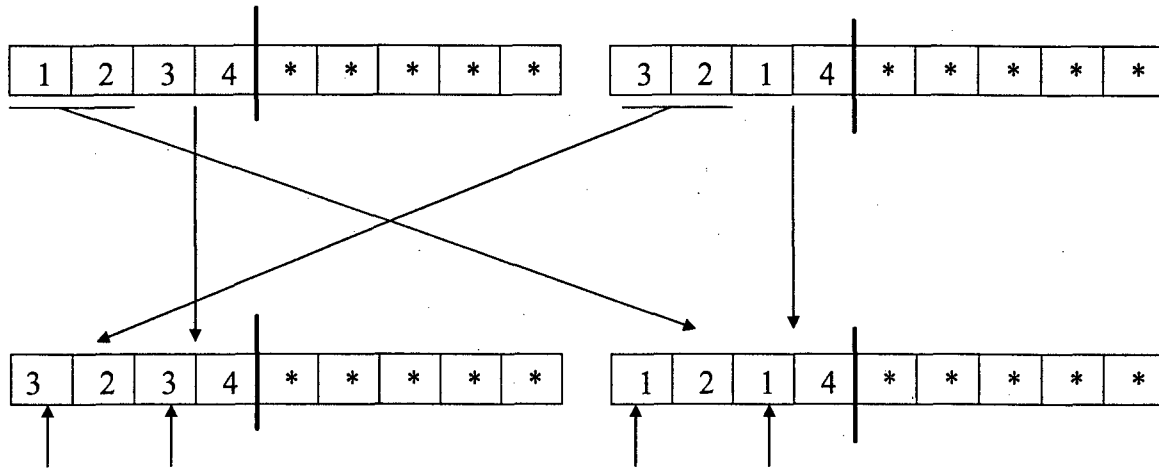


Figure 15: Step One of Crossover Routine

Notice that in each of the new chromosomes, a mission assignment is duplicated. Specifically, the chromosome on the right has two systems assigned to mission three and none to mission one, and the chromosome on the left has two systems assigned to mission one and none to mission three. To correct this situation, a duplicate assignment gene is chosen from each chromosome. The value of that gene is initialized again. The chromosome is checked again for duplicate missions, and the above process is repeated until no duplicate mission assignments exist.

Next, the remaining $q*m*h$ genes undergo crossover. The position and length of the crossover are also chosen randomly. Again, consider the chromosomes shown in *Figure 14*. Suppose the position of the crossover is gene 2 (in the second section of the chromosome) and the length of the crossover is 3. *Figure 16* shows the second step in the crossover routine.

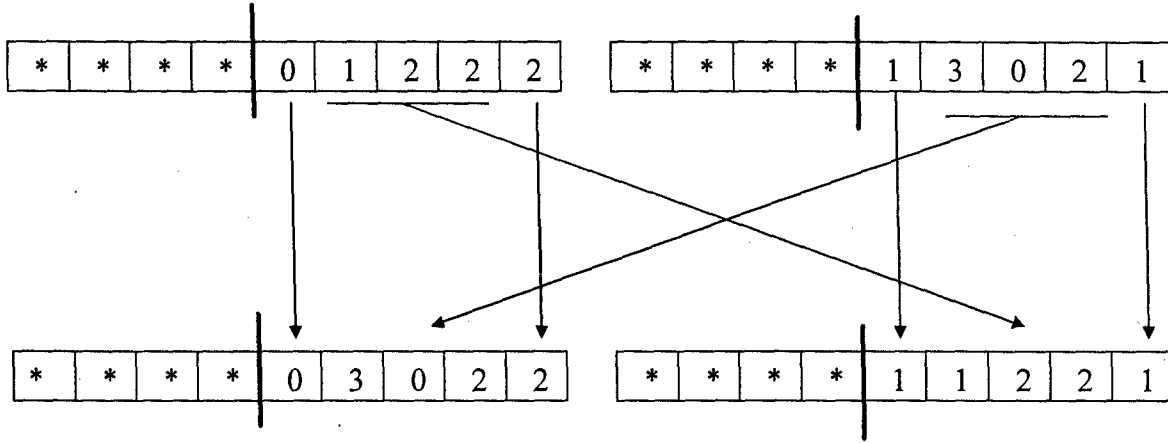


Figure 16: Step Two of Crossover Routine

For each chromosome created in the crossover routine, the probability of mutation is defined to be 0.1. That is, each chromosome has a 10% chance of undergoing mutation. Once crossover and mutation are complete, the parents and children are merged to create a new generation.

The new generation is ranked according to overall reliability. The best solutions are subjected to crossover, and chromosomes resulting from crossover are subjected to mutation. The parents and children are then merged forming a new generation. This process is repeated for a user-specified number of generations.

As a numerical example, consider the dynamic mission profile shown in *Figure 17*. In this scenario, we consider the set of systems shown in *Figure 13* have $q = 4$ and $m = 3$. We consider $s = 2$ limited maintenance resources, specifically, $\beta_1 = 8$ and $\beta_2 = 6$. The next mission durations (in hours) are $t_1 = 4$, $t_2 = 1.75$, $t_3 = 2$, $t_4 = 0.5$, and the remaining system parameters are shown in *Table 5.1*. Note that system 1 returns from its previous mission at the beginning of time bucket 1. Systems 2 and 4 will return at the beginning of time bucket 3, and system 3 will return at the beginning of time bucket 4. Further, note that upcoming missions 1 and 2 depart at the beginning of time bucket 4, mission 3 at the beginning of time bucket 5, and mission 4 at the beginning of time bucket 6. This implies that $ST^{\max} = 6$.

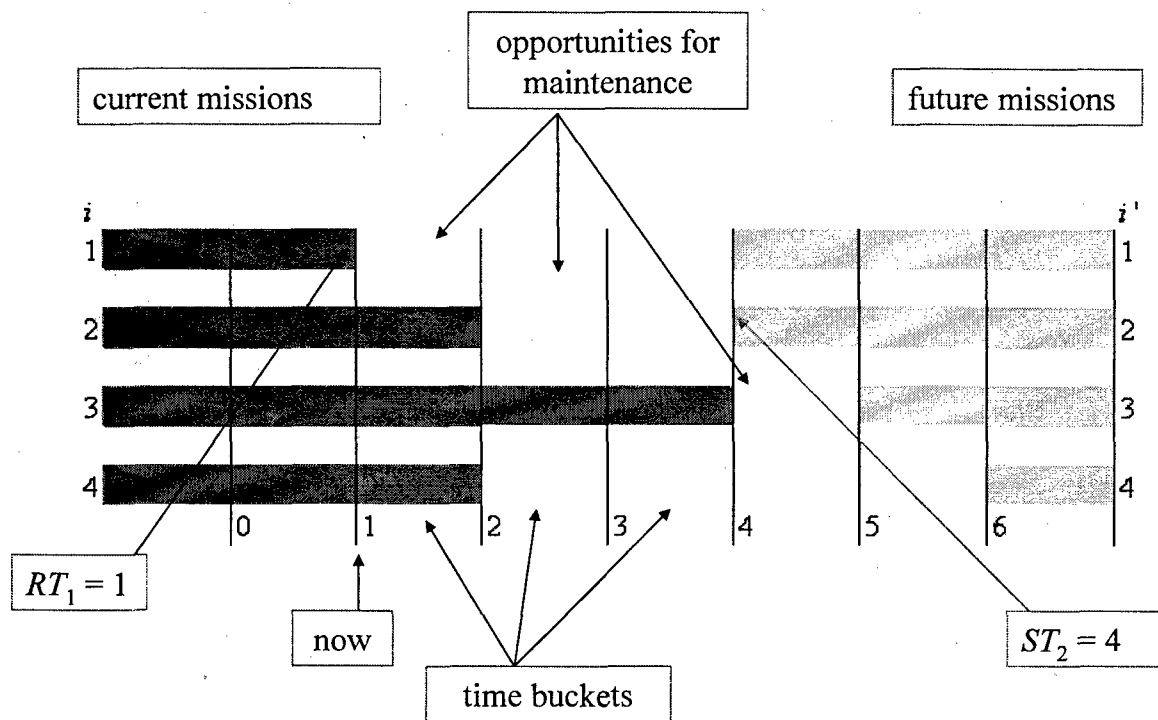


Figure 17: Example Dynamic Mission Profile

Table 5.1: Model Parameters for Example Set of Systems

Subsystem	λ_j (in failures per hour)	α_{j1}	α_{j2}
1	0.10	2	3
2	0.16	3	1
3	0.06	2	2

At the beginning of time bucket 1, “now,” system 1 returns from its mission. This system has three failed components as shown in *Figure 18*. At this point, the number of failed components in the remaining systems are just estimates.

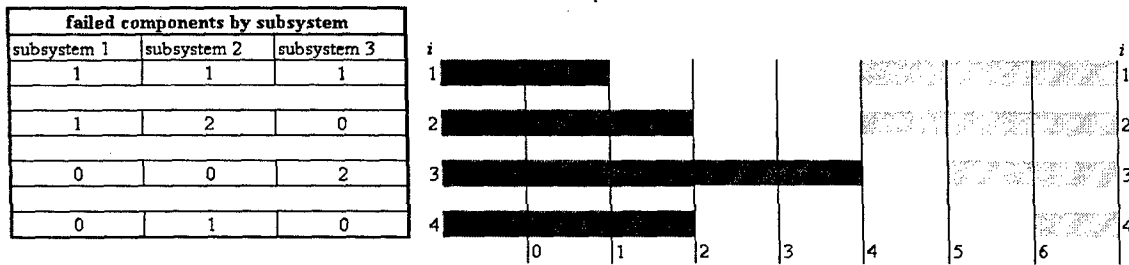


Figure 18: Dynamic Mission Profile with Component Failures

We can use our selective maintenance model to determine the mission assignment for each system and which components to repair in each time bucket to maximize the overall reliability. We solve this scenario using the GA described in *Section 5.4*. This approach uses an application developed in Visual Basic® and programmed as a macro within an Excel spreadsheet. Instructions for using the spreadsheet appear in *Appendix C*. The model outputs are shown in *Figure 19*; specifically, system 1 is assigned to upcoming mission 2. It will have a component in subsystem 1 repaired in time bucket 1 and a component in subsystem 3 repaired in time bucket 2. System 2 is assigned to upcoming mission 3. It will have a component in subsystem 1 repaired in time bucket 3 and a component in subsystem 2 repaired in time bucket 4. System 3 is assigned to upcoming mission 4. It will have one component in subsystem 3 repaired in time bucket 4 and another one in time bucket 5. Finally, system 4 is assigned upcoming mission 1 and will have a component in subsystem 2 repaired in time bucket 2. The overall reliability of this solution is 0.5225.

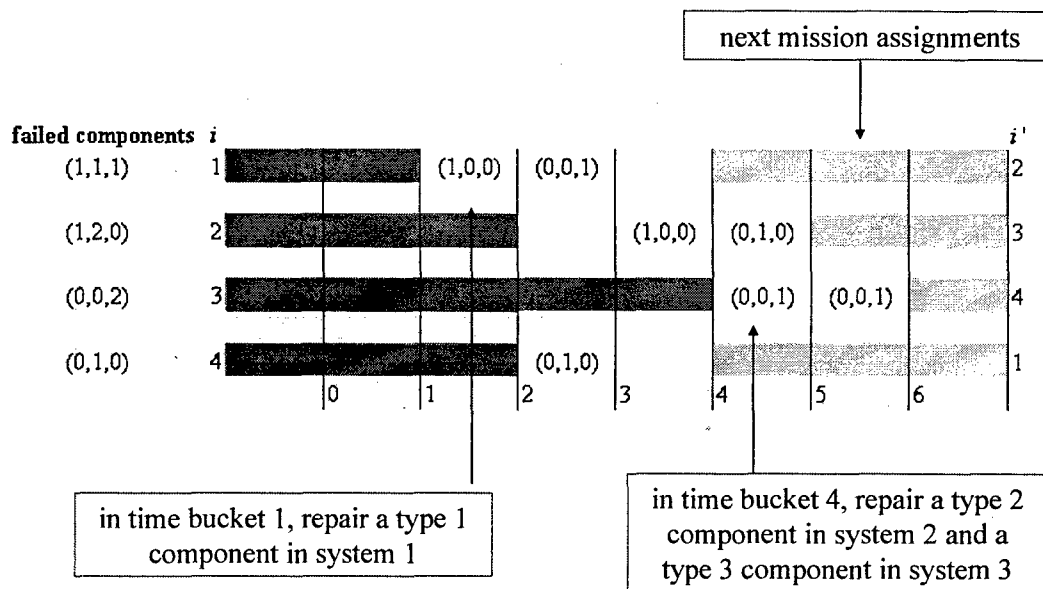


Figure 19: Solution for Numerical Example

It is important to note that the solution presented was run at the beginning of time bucket 1, and the number of failed components in systems 2, 3, and 4 were estimates. At the beginning of time bucket 2, the number of failed components in systems 2 and 4 will be known. Therefore, it may be beneficial to enter the updated information into the model and run it again.

6 Future Opportunities

Vast opportunities for extending this research exist. For example, all three of the models presented could be extended to include a variable cost of maintenance. The model's objective function could be modified to minimize the cost associated with maintenance actions; however, such a modification would require a penalty value associated with failed missions. Another extension could include the ability to cancel a mission if its probability of success is below some threshold.

The selective maintenance models presented in this research treat decision-making relative to a single future mission. These models could be extended to capture the reliability and performance of the set of systems over multiple missions. If a system is required to perform a sequence of missions, then the selective maintenance decisions directly affect system reliability for the next mission and indirectly affect the system reliability for later missions. This multiple-mission selective maintenance issue will be addressed in Applied Logistics Research Project BSIT0204. The models presented in this research will serve as a foundation for the upcoming project.

Also, the models presented in this research consider components, subsystems, and systems that are in one of two states: functioning or failed. It may be more realistic to classify component status using more than two discrete levels. Additionally, the performance of a system can be evaluated using several measures. Therefore, another potential extension to this research includes considering multiple states of component function and multiple performance measures for a set of systems.

Currently, issues surrounding cannibalization are being investigated in TLI-Military Logistics research project MM0206. Perhaps in future efforts, selective maintenance modeling could be incorporated into cannibalization decisions.

Finally, we would like to develop a USAF-specific example to apply the models developed in this research and the research conducted in the other projects mentioned above. Through continuous collaboration with our partners at the Air Force Research Laboratory (AFRL), we hope to identify a subsystem in an aircraft or a subsystem within a subsystem such that selective maintenance decisions can be evaluated using the models developed in this research.

References

1. Abara, J. (1989). "Applying Integer Linear Programming to the Fleet Assignment Problem," *Interfaces*, Vol. 19, No. 4, pp. 20-28.
2. Barnhart, C., N.L. Boland, L.W. Clarke, E.L. Johnson, G.L. Nemhauser, and R.G. Shenoi (1989). "Flight String Models for Aircraft Fleeting and Routing," *Transportation Science*, Vol. 32, No. 3, pp. 208-220.
3. Cassady, C.R., W.P. Murdock and E.A. Pohl (2001). "A Deterministic Selective Maintenance Model for Complex Systems," *Recent Advances in Reliability and Quality Engineering* (H. Pham, Editor), World Scientific, Singapore, pp. 311-325.
4. Cassady, C.R., E.A. Pohl and W.P. Murdock (2001). "Selective Maintenance Modeling for Industrial Systems," *Journal of Quality in Maintenance Engineering*, Vol. 7, No. 2, pp. 104-117.
5. Cassady, C.R., W.P. Murdock and E.A. Pohl (2001). "Selective Maintenance for Support Equipment Involving Multiple Maintenance Actions," *European Journal of Operational Research*, Vol. 129, No. 2, pp. 252-258.
6. Cassady, C.R. and K. Schneider (2003). "Fleet-Level Selective Maintenance Modeling," Working Paper.
7. Chen, C., M. Q-H. Meng, and M.J. Zuo (1999). "Selective Maintenance Optimization for Multi-State Systems," *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*.
8. Cho, D.I. and M. Parlar (1991). "A Survey of Maintenance Models for Multi-Unit Systems," *European Journal of Operational Research*, Vol. 51, pp. 1-23
9. Clarke, L.W., C.A. Hane, E.L. Johnson, and G.L. Nemhauser (1996). "Maintenance and Crew Considerations in Fleet Assignment," *Transportation Science*, Vol. 30, pp. 249-260.

10. Dekker, R. (1996). "Applications of Maintenance Optimization Models: A Review and Analysis," *Reliability Engineering and System Safety*, Vol. 51, pp. 229-240.
11. Feo, T.A. and J.F. Bard (1989). "Flight Scheduling and Maintenance Base Planning," *Management Science*, Vol. 35, No. 12, pp. 1415-1432.
12. Hane, C.A., C. Barnhart, E.L. Johnson, R.E. Marsten, G.L. Nemhauser, and G. Sigismondi (1995). "The Fleet Assignment Problem: Solving A Larger-Scale Integer Program," *Mathematical Programming*, Vol. 70, pp. 211-232.
13. Lam, M. (1995), "An Introduction to Airline Maintenance," *The Handbook of Airline Economics*, pp. 397-406.
14. McCall, J.J. (1965). "Maintenance Policies for Stochastically Failing Equipment: A Survey," *Management Science*, Vol. 11, pp. 493-524.
15. Pierskalla, W.P. and J.A. Voelker (1976). "A Survey of Maintenance Models: The Control and Surveillance of Deteriorating Systems," *Naval Research Logistics Quarterly*, Vol. 23, pp. 353-388.
16. Rice, W.F., C.R. Cassady and J.A. Nachlas (1998) "Optimal Maintenance Plans under Limited Maintenance Time," *Industrial Engineering Research '98 Conference Proceedings*.
17. Rushmeier, R.A. and S.A. Kontogiorgis (1997). "Advances in the Optimization of Airline Fleet Assignment," *Transportation Science*, Vol. 31, pp. 159-169.
18. Sherif, Y.S. and M.L. Smith (1981). "Optimal Maintenance Models for Systems Subject to Failure – A Review," *Naval Research Logistics Quarterly*, Vol. 28, pp. 47-74.
19. Valdez-Florez, C. and R. Feldman (1989). "Survey of Preventive Maintenance Models for Stochastically Deteriorating Single-Unit Systems," *Naval Research Logistics*, Vol. 36, pp. 419-446.
20. Wang, H. (2002). "A Survey of Maintenance Policies of Deteriorating Systems," *European Journal of Operational Research*, Vol. 139, pp. 469-489.

Appendix A: Spreadsheet Instructions for Evaluating Model P

Total Enumeration Strategy

File name: Total Enumeration_ModelP

Setting up the spreadsheet for use

When you open the file, you may receive a message similar to the one shown in *Figure A 1*.

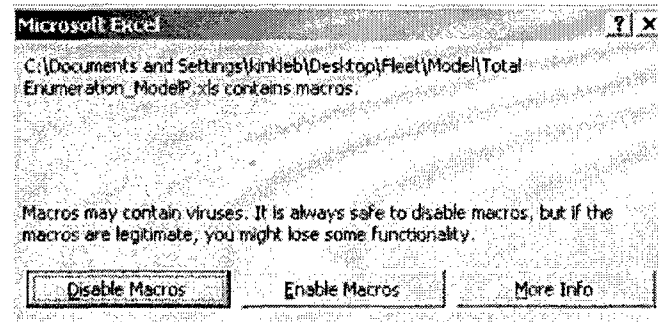


Figure A 1: Macro Notification

All the Visual Basic[®] code used to evaluate the model is written within a macro. Therefore, you should click **Enable Macros**. This will open the Inputs worksheet shown in *Figure A 2*. Next, fill in the number of systems, q , and subsystems, m , in the scenario you wish to evaluate. Then, determine the number of limited maintenance resources, s . (Note that for problems involving more than 4 systems, computation times become extremely long. Therefore, large scenarios should be evaluated using the GA.) Once you have entered these values, activate a blank cell (by clicking it), and click the "Reset Fields" button. *Figure A 3* shows the updated Inputs worksheet for the scenario in which $q = 2$, $m = 3$, and $s = 2$.

Microsoft Excel - Total enumeration_model.xls

File Edit View Insert Format Tools Data Window Help

Times New Roman 12 B / U

Σ ∑ ∑ ∑ 85%

Reply with Changes... Find Proxies...

F26

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		72 Total Solutions														
2		0:00:00														
3																
4		Number of Systems		q	0											
5		Number of Subsystems		m	0											
6		Number of limited resources		s	0											
7																
8		Number of failed components in each subsystem - h_j														
9																
10																
11																
12		Subsystem		r_j	n_j											
13																
14																
15																
16		Amount of resource consumed to repair a component within a subsystem - α_{ij}														
17																
18																
19		Amount of each resource available - β_j		β_j												
20																
21		Output														
22																
23																
24																

Ready

NUM

Instructions
 Step 1 Enter q, m, and s values
 Step 2 Press "Reset Fields"
 Step 3 Enter data into input fields
 Step 4 Press "Go"

New
Go

Reset Fields

Figure A 2: Inputs Worksheet

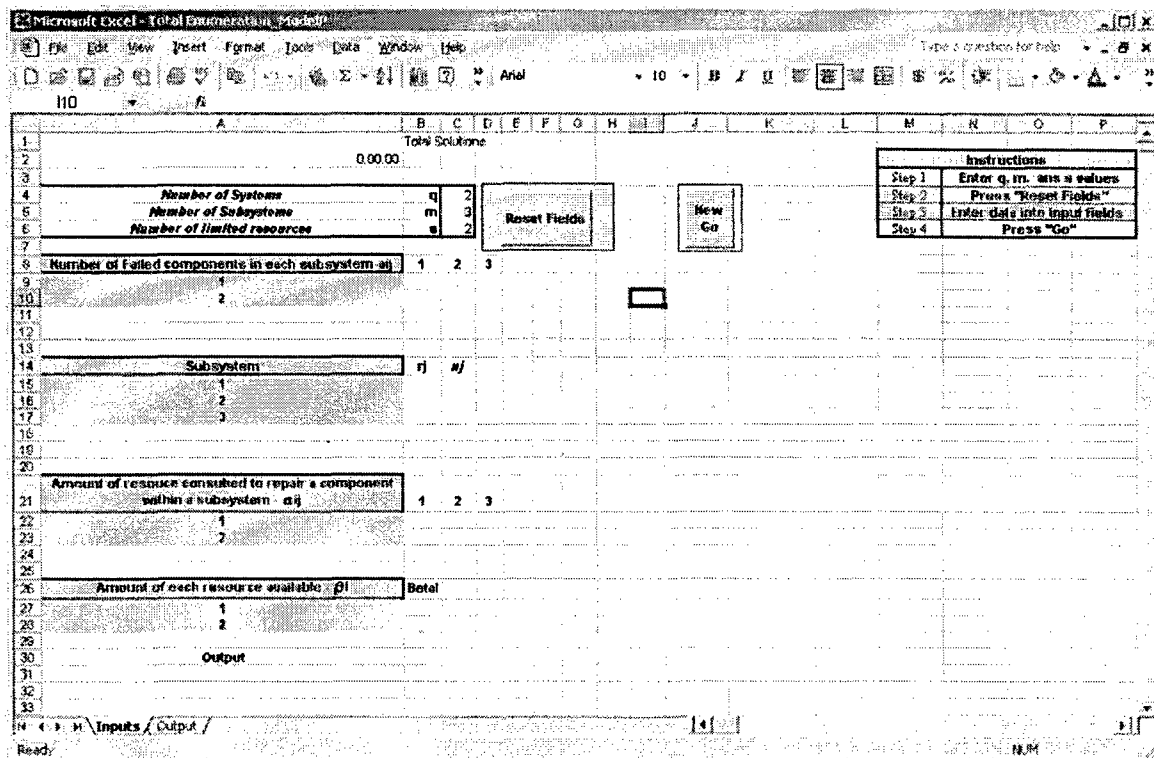


Figure A 3: Updated Inputs Worksheet

Entering model parameter values

For each scenario you wish to evaluate, you must enter the model parameters. The first parameters of interest are the reliabilities of components in each subsystem, r_j , and the number of independent and identical copies of components in each subsystem, n_j . These values are entered as shown in *Figure A 4*.

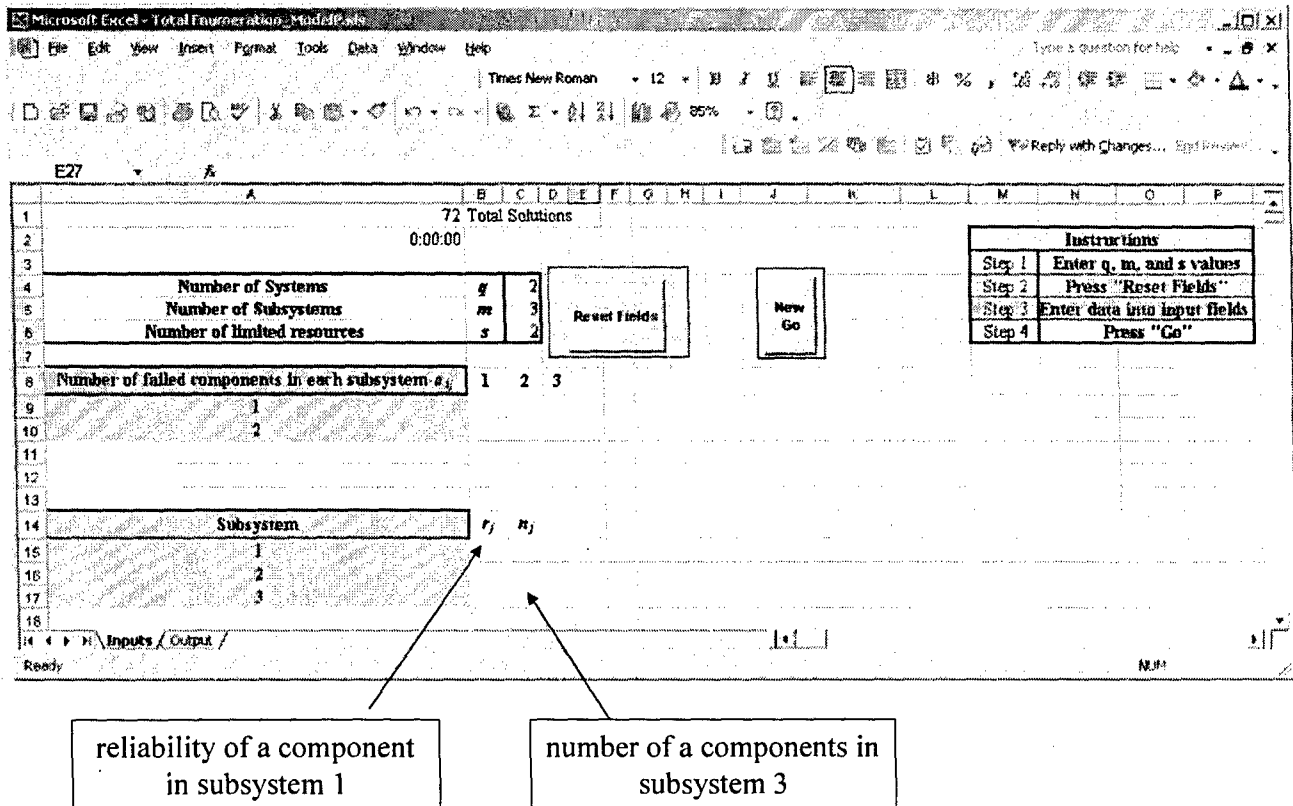


Figure A 4: Input Parameters for Component Reliability and Number of Components

Next, you must enter the amount of resource consumed by repairing a type j component, α_{jl} and the amount of each resource available, β_l . These values must be entered for each of the s limited maintenance resources (in this case, $s = 2$) as shown in *Figure A 5*.

Microsoft Excel - Fatal Environment model.xls

File Edit View Insert Format Tools Data Window Help

Times New Roman 10 05%

E34

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
4	Number of Systems	q	2											Step 2	Press "Reset Fields"	
5	Number of Subsystems	m	3											Step 3	Enter data into input fields	
6	Number of limited resources	s	2											Step 4	Press "Go"	
7																
8	Number of failed components in each subsystem - n_j		1	2	3											
9	1															
10	2															
11																
12																
13																
14	Subsystem		r_j	n_j												
15	1															
16	2															
17	3															
18																
19																
20																
21	Amount of resource consumed to repair a component within a subsystem - α_{ij}		1	2	3											
22	1															
23	2															
24																
25																
26	Amount of each resource available - β_i		β_i													
27	1															
28	2															
29																
30	Output															
31																

Ready

amount of resource 1 consumed by repairing a component in subsystem 1

amount of resource 2 consumed by repairing a component in subsystem 3

amount of resource 2 available

Figure A 5: Input Parameters for Limited Maintenance Resources

Consider the numerical example from *Section 3*. Recall that $q = 2$, $m = 3$, and $s = 2$, and note that $\beta_1 = 16$ and $\beta_2 = 10$. The remaining model parameters are shown in *Table A 1*. *Figure A 6* provides a snapshot of the Inputs worksheet with the appropriate parameter inputs.

Table A 1: Model Parameters for Example Set of Systems

Subsystem	n_j	r_j	α_{11}	α_{12}
1	2	0.90	2	3
2	3	0.05	3	1
3	2	0.94	2	2

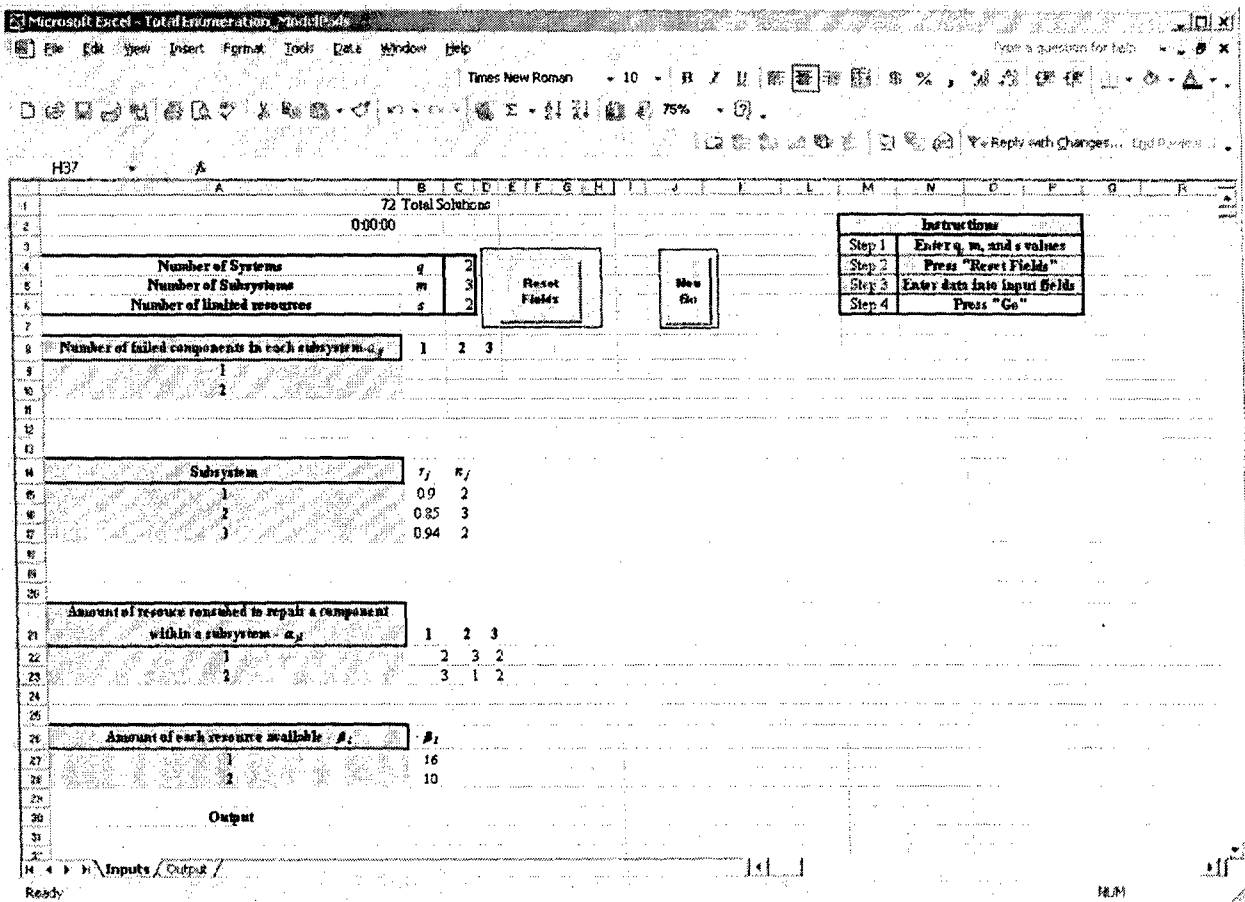


Figure A 6: Inputs Worksheet with Example Parameter Values

Running an experiment

Once you have entered all the model parameters, you are now ready to run an experiment. Determine the number of failed components in each subsystem. If we again consider the numerical example from Section 3, the failed components for this scenario are shown in Table A 2. These values are entered into the spreadsheet in the same manner as the other model parameters.

Table A 2: Failed Components for Example Set of Systems

System (q)	Subsystem (i)		
	1	2	3
1	$a_{11} = 1$	$a_{12} = 2$	$a_{13} = 1$
2	$a_{21} = 0$	$a_{22} = 2$	$a_{23} = 1$

To run the model, simply activate a blank cell and click **New Go**. You will be shown the total number of feasible solutions evaluated, the run time, and the optimal solution in *Figure A 7*. For this scenario, 72 solutions were evaluated in less than one hundredth of a second. The optimal solution is to repair one component in each subsystem of system 1 (i.e. $d_{11} = d_{12} = d_{13} = 1$) and one component in subsystems 2 and 3 of system 2 (i.e. $d_{21} = 0, d_{22} = d_{23} = 1$). The resulting overall reliability is 0.9479. The solution of the model is also shown on the Output worksheet as shown in *Figure A 8*.

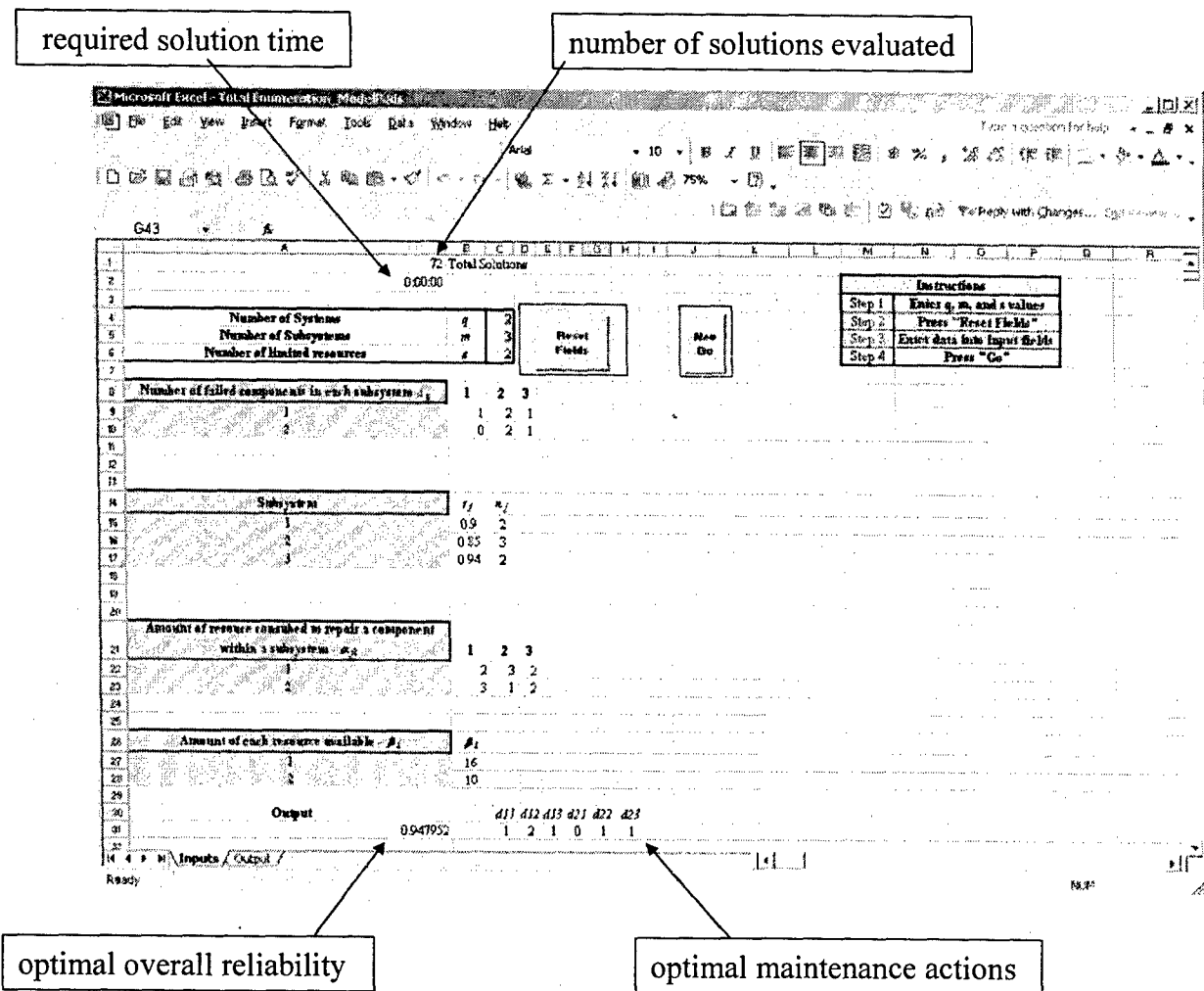


Figure A 7: Model Output

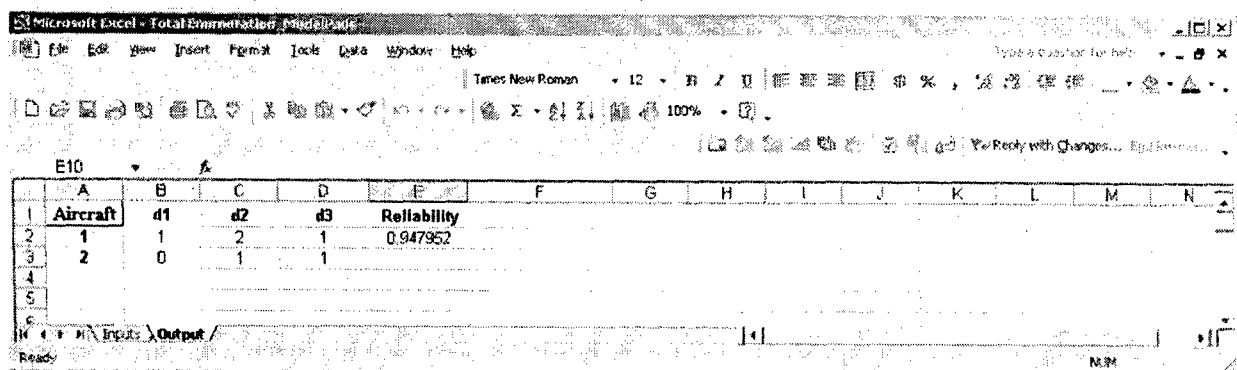


Figure A 8: Output Worksheet

Genetic Algorithm

File name: GA_ModelP

Setting up the spreadsheet

When you open the file, you may receive a message similar to the one shown in *Figure A 9*.

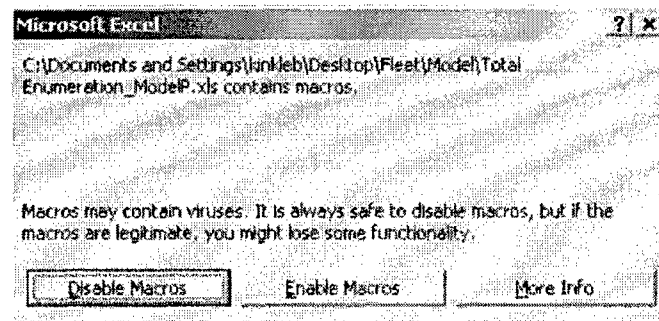


Figure A 9: Macro Notification

All the Visual Basic[®] code used to evaluate the model is written within a macro. Therefore, you should click **Enable Macros**. This will open the GA Settings worksheet with the default settings. Recall that population size is the number of candidate solutions or chromosomes that the GA monitors, and the number of generations is the number of times the population undergoes crossover and mutation. Finally, the seed value relates to the random number generator. This cell may have a value from 1 to 100. This worksheet is shown in *Figure A 10*.

Next, you must enter the parameters of the scenario you wish to evaluate. This is accomplished by entering data into the Inputs Worksheet. Fill in the number of systems, q , and subsystems, m , in the scenario you wish to evaluate. Then, determine the number of limited maintenance resources, s . Once you have entered these values, activate a blank cell (by clicking it), and click **Reset Fields**. *Figure A 11* shows the Inputs worksheet for the scenario in which $q = 2$, $m = 3$, and $s = 2$.

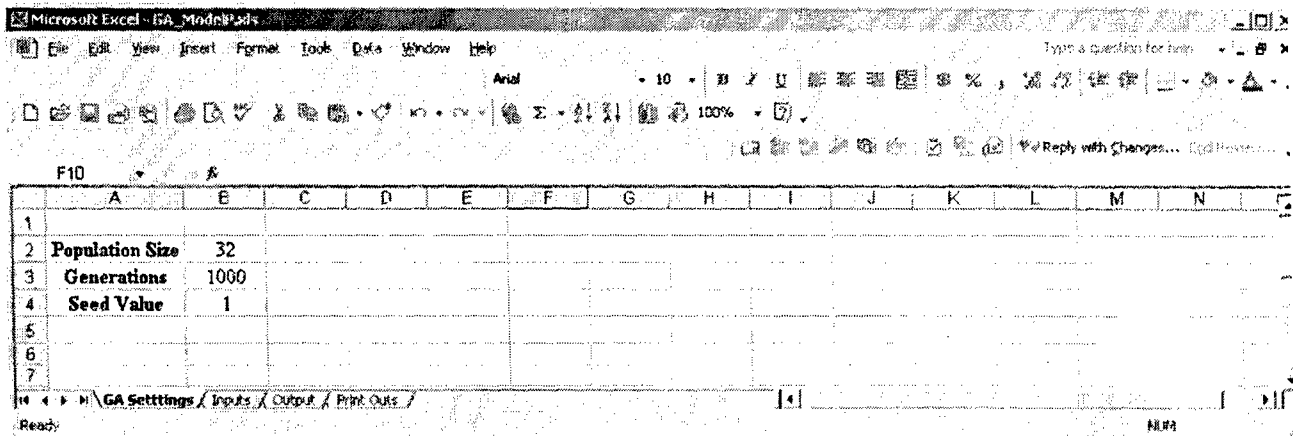


Figure A 10: GA Settings Worksheet

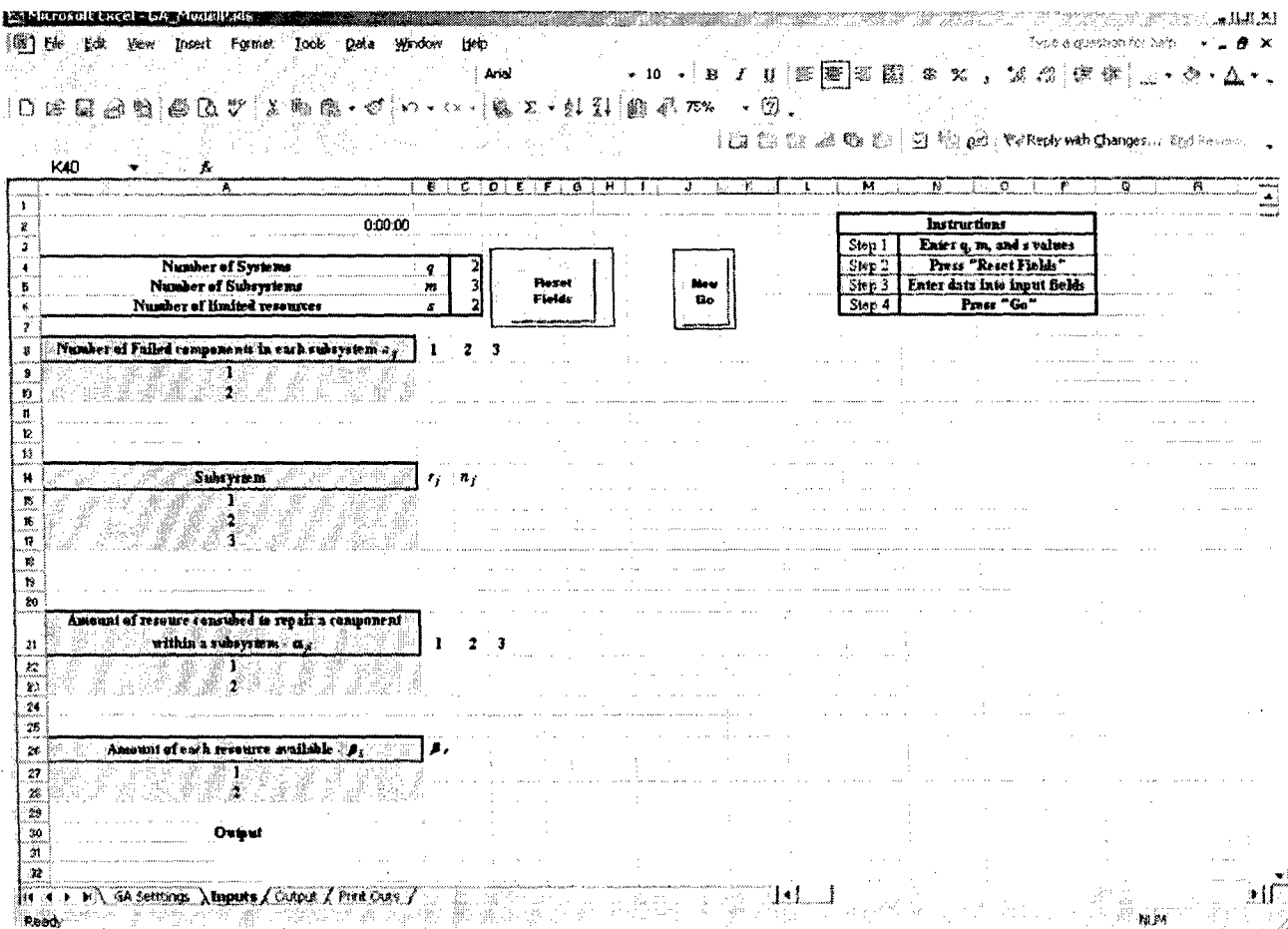


Figure A 11: Input Worksheet with Example Parameter Inputs

Entering model parameter values

For each scenario you wish to evaluate, you must enter the model parameters. The parameters of interest are the reliabilities of components in each subsystem, r_j , the number of independent and identical copies of components in each subsystem, n_j , the amount of resource consumed by repairing a type j component, α_{ji} , and the amount of each resource available, β_i . These values must be entered for each of the s limited maintenance resources (in this case, $s = 2$) as shown in *Figure A 12*.

Consider the numerical example from *Section 3*. Recall that $q = 2$, $m = 3$, and $s = 2$, and note that $\beta_1 = 16$ and $\beta_2 = 10$. The remaining model parameters are shown in *Table A 3*. *Figure A 12* provides a snapshot of the Inputs worksheet with the appropriate parameter inputs.

Table A 3: Model Parameters for Example Set of Systems

Subsystem	n_j	r_j	α_{j1}	α_{j2}
1	2	0.90	2	3
2	3	0.85	3	1
3	2	0.94	2	2

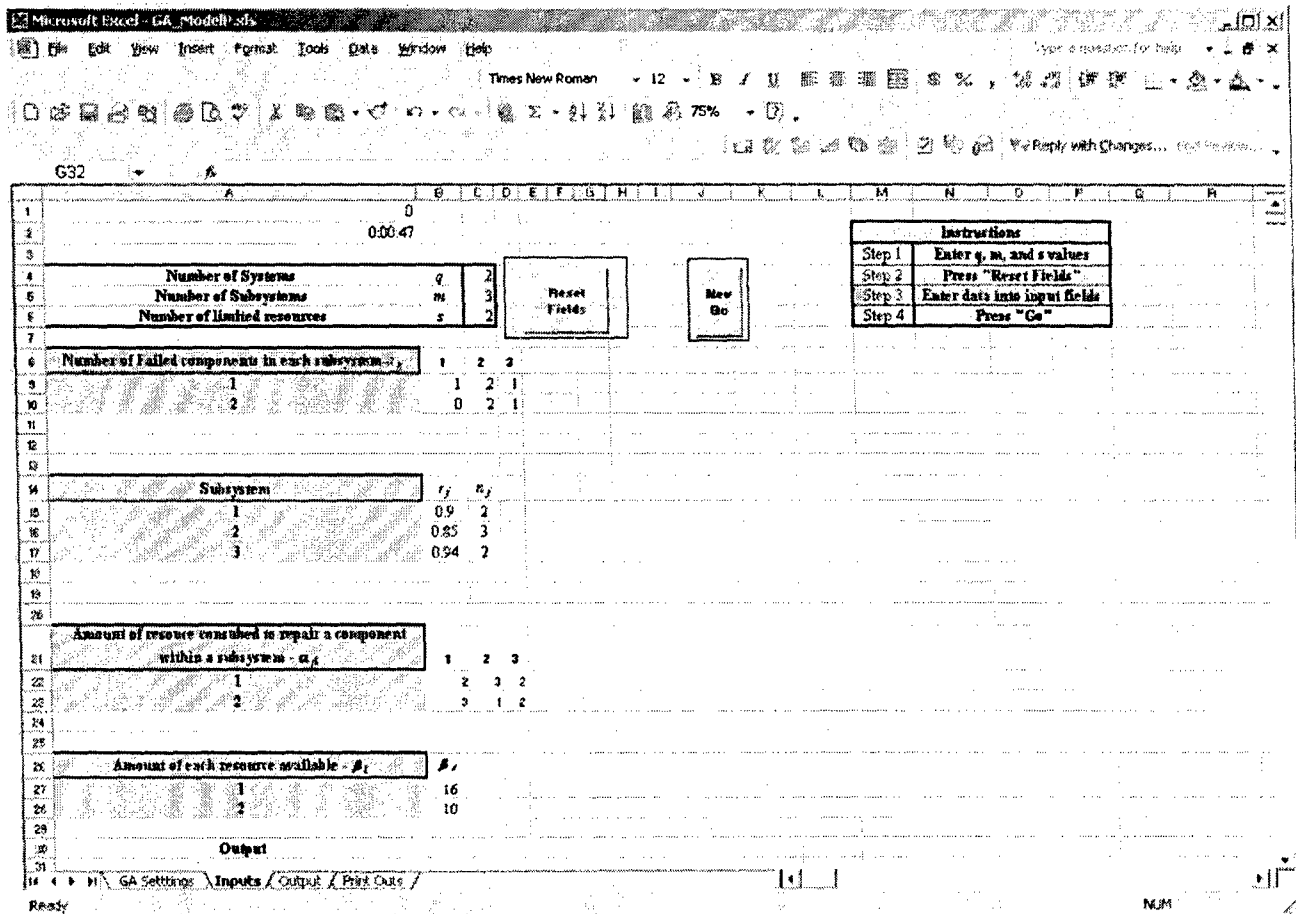


Figure A 12: Inputs Worksheet with Example Parameters

Running an experiment

Once you have entered all the model parameters, you are now ready to run an experiment. Determine the number of failed components in each subsystem. If we again consider the numerical example from Section 3, the failed components for this scenario are shown in Table A 4. These values are entered into the spreadsheet in the same manner as the other model parameters.

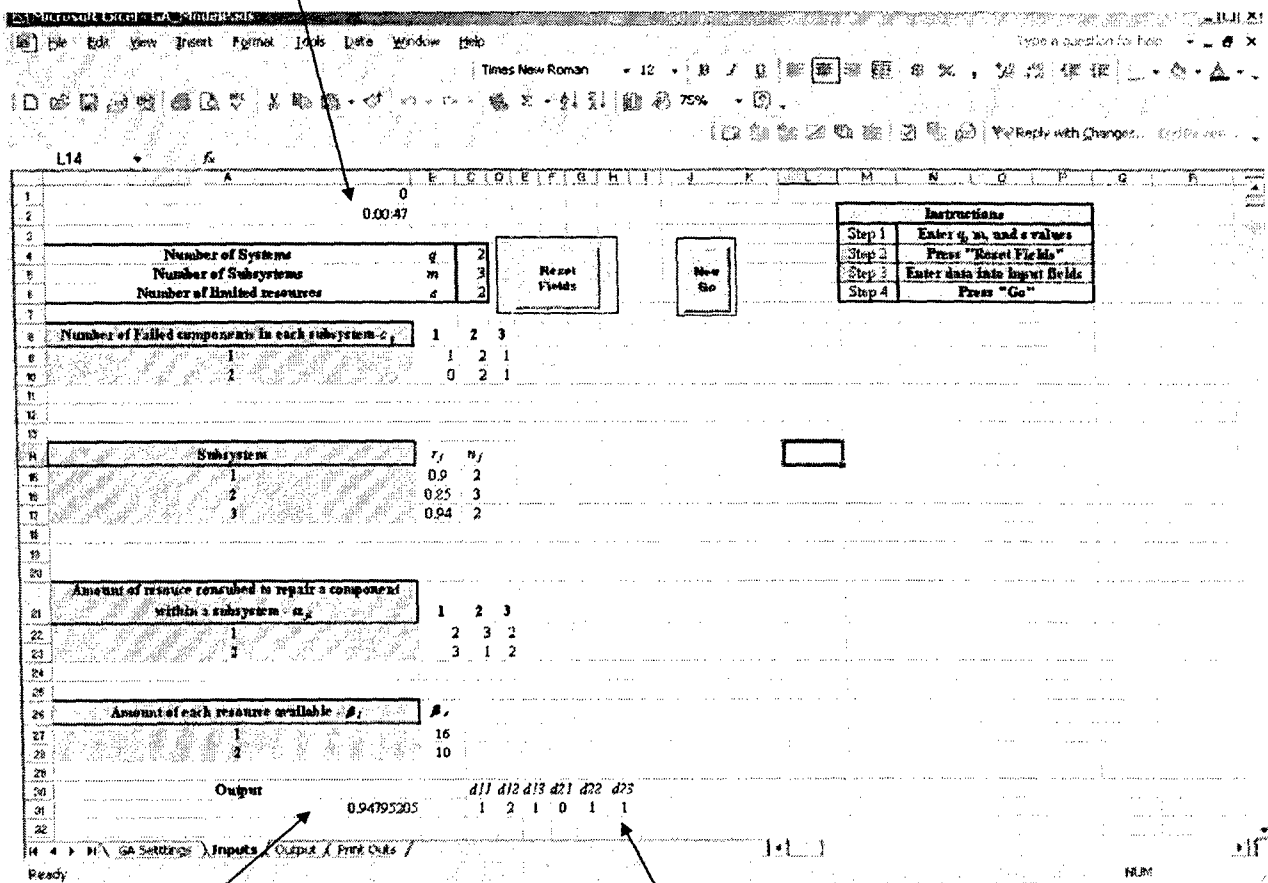
Table A 4: Failed Components for Example Set of Systems

System (q)	Subsystem (i)		
	1	2	3
1	$a_{11} = 1$	$a_{12} = 2$	$a_{13} = 1$

System (q)	Subsystem (i)		
	1	2	3
2	$a_{21} = 0$	$a_{22} = 2$	$a_{23} = 1$

To run the model, simply activate a blank cell and click **New Go**. You will be shown the total number of feasible solutions evaluated, the run time, and the optimal solution, as in *Figure A 13*. For this scenario, the GA ran for 47 seconds and provided the same solution as the total enumeration approach. Therefore, the optimal solution is to repair one component in each subsystem of system 1 (that is, $d_{11} = d_{12} = d_{13} = 1$) and one component in subsystems 2 and 3 of system 2 (specifically, $d_{21} = 0$, $d_{22} = d_{23} = 1$). The resulting overall reliability is 0.9479.

required solution time



optimal overall reliability

optimal maintenance actions

Figure A 13: Model Output

Appendix B: Spreadsheet Instructions for Evaluating Model SP

File name: GA_ModelSP

Setting up the spreadsheet

When you open the file, you may receive a message similar to the one shown in *Figure B 1*.

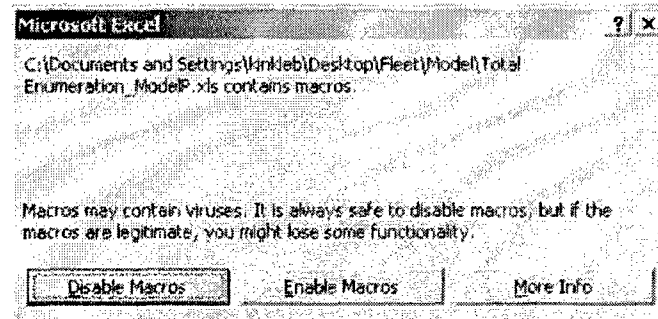


Figure B 1: Macro Notification

All the Visual Basic[®] code used to evaluate the model is written within a macro. Therefore, you should click **Enable Macros**. This will open the GA Settings worksheet with the default settings. Recall that population size is the number of candidate solutions or chromosomes the GA monitors, and the number of generations is the number of times the population undergoes crossover and mutation. Finally, the seed value relates to the random number generator. This cell may have a value from 1 to 100. This worksheet is shown in *Figure B 2*.

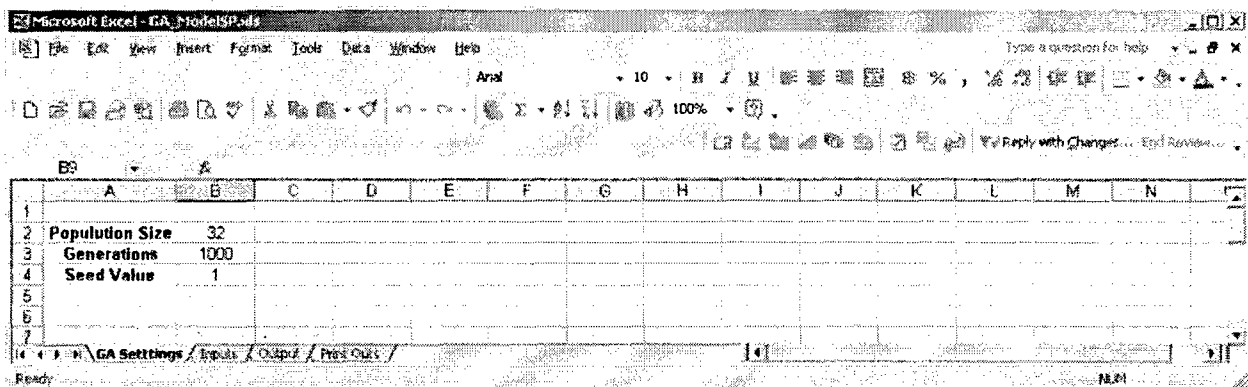


Figure B 2: GA Settings Worksheet

Next, you must enter the parameters of the scenario you wish to evaluate. This is accomplished by entering data into the Inputs Worksheet. Fill in the number of systems, q , and subsystems, m , in the scenario you wish to evaluate. Then, determine the number of limited maintenance resources, s . Once you have entered these values, activate a blank cell (by clicking it), and click **Reset Fields**.

Entering model parameter values

For each scenario you wish to evaluate, you must enter the model parameters. The parameters of interest are the failure rates of components in each subsystem, λ_j , the number of independent and identical copies of components in each subsystem, n_j , the amount of resource consumed by repairing a type j component, α_{j1} , and the amount of each resource available, β_1 . These values must be entered for each of the s limited maintenance resources (in this case, $s = 2$) as shown in *Figure B 3*.

Consider the numerical example from *Section 4*. Recall that $q = 2$, $m = 3$, and $s = 2$, and note that $\beta_1 = 16$ and $\beta_2 = 14$. The next mission durations (in hours) are $t_1 = 4$, $t_2 = 1.75$, $t_3 = 2$, $t_4 = 0.5$, and the remaining model parameters are shown in *Table B 1*. *Figure B 3* provides a snapshot of the Inputs worksheet with the appropriate parameter inputs.

Table B 1: Model Parameters for Example Set of Systems

Subsystem	n_j	λ_j (in failures per hour)	α_{j1}	α_{j2}
1	2	0.10	2	3
2	3	0.16	3	1
3	2	0.06	2	2

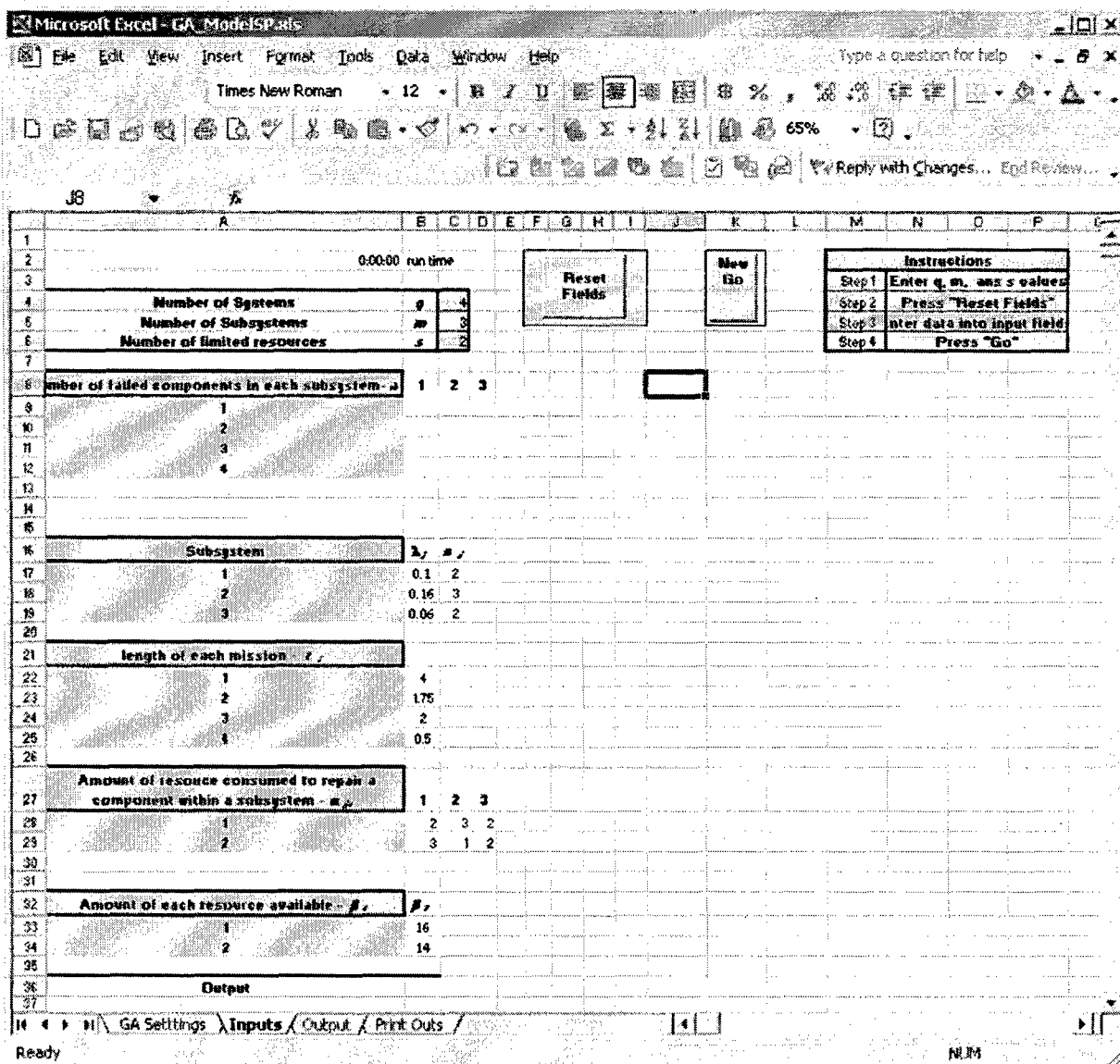


Figure B 3: Inputs Worksheet with Example Parameters

Running an experiment

Once you have input all of the model parameters, you are now ready to run an experiment. Determine the number of failed components in each subsystem. If we again consider the numerical example from *Section 4*, the failed components for this scenario are shown in *Table B 2*. These values are entered into the spreadsheet in the same manner as the other model parameters.

Table B 2: Failed Components for Example Set of Systems

System (q)	Subsystem (i)		
	1	2	3
1	$a_{11} = 1$	$a_{12} = 1$	$a_{13} = 1$
2	$a_{21} = 1$	$a_{22} = 2$	$a_{23} = 0$
3	$a_{31} = 1$	$a_{32} = 2$	$a_{33} = 1$
4	$a_{41} = 0$	$a_{42} = 0$	$a_{43} = 1$

To run the model, simply activate a blank cell and click **New Go**. For this scenario, the GA ran for one minute and eight seconds, and the output shown on the first line of the Print Outs worksheet. As you can see in *Figure B 1*, the GA identified two different solutions having an overall reliability of 0.5347.

Microsoft Excel - GA_ModelSP.xls

File Edit View Insert Format Tools Data Window Help

True a question for help

Arial 10

Reply with Changes...

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1						d11	d12	d13	d21	d22	d23	d31	d32	d33	d41	d42	d43		
2	1	3	4	2	1	1	1	1	1	1	0	0	0	1	0	0	1	0.53472	
3	3	1	4	2	1	0	1	1	1	2	0	0	0	1	0	0	1	0.53472	
4	3	1	4	2	1	0	1	1	1	2	0	0	0	0	0	0	1	0.57036	
5	1	3	4	2	1	1	1	1	1	1	0	0	0	0	0	0	1	0.57036	
6	1	3	4	2	1	1	1	1	1	1	0	0	0	0	0	0	1	0.57036	
7	1	3	4	2	1	0	1	1	1	2	0	0	0	1	0	0	0	0.59662	
8	1	3	4	2	1	1	1	1	1	1	0	0	0	1	0	0	0	0.59662	
9	3	1	4	2	1	0	0	1	1	2	0	0	0	1	0	0	1	0.59662	
10	3	1	4	2	1	1	1	1	1	2	0	0	0	0	0	0	0	0.59662	
11	1	3	4	2	1	0	1	1	1	1	0	0	0	1	1	0	0	0.49534	
12	2	1	4	2	1	0	1	1	1	2	0	1	0	0	0	0	0	0.49534	
13	1	2	4	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0.49534	
14	1	3	4	2	1	1	1	1	1	1	0	0	0	0	0	1	0	0.49534	
15	2	1	4	2	1	0	1	1	1	2	0	0	0	0	0	0	0	0.49534	
16	1	3	4	2	1	1	1	1	1	1	0	0	0	0	0	1	0	0.49534	
17	1	3	4	2	1	1	1	0	1	2	0	0	0	0	0	0	1	0.49534	
18	1	3	4	2	1	1	1	1	1	1	0	0	0	1	0	0	0	0.49534	
19	1	3	4	2	1	0	1	1	1	1	0	0	0	1	0	0	0	0.49534	
20	1	3	4	2	1	0	1	1	1	2	0	0	0	0	0	0	0	0.49534	
21	3	1	4	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0.49534	
22	2	1	4	2	1	0	1	1	1	2	0	0	0	0	0	0	0	0.49534	
23	3	1	4	2	1	1	0	1	1	2	0	0	0	0	0	0	0	0.49534	
24	2	1	4	2	1	0	0	1	1	2	0	0	1	0	0	0	0	0.49534	
25	3	1	4	2	1	1	0	1	1	2	0	0	0	0	0	0	0	0.49534	
26	1	3	4	2	1	0	1	1	1	1	0	0	0	2	0	0	0	0.49534	
27	3	1	4	2	1	1	1	1	1	1	0	0	0	1	0	0	0	0.49534	
28	3	1	4	2	1	0	0	1	1	2	0	0	0	0	0	0	0	0.49534	
29	1	3	4	2	1	0	1	0	1	1	0	0	0	1	0	0	0	0.49534	
30	3	1	4	2	1	0	1	1	0	2	0	0	0	0	0	0	0	0.49534	
31	1	3	4	2	1	0	1	1	1	0	0	0	0	1	0	0	0	0.49534	
32	3	1	4	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0.49534	
33	1	3	4	2	1	0	0	1	0	1	0	0	0	0	0	0	0	0.2915	

GA settings Inputs Output Print Outs

Ready

N:2

Figure B 4: Model Output

Appendix C: Spreadsheet Instructions for Evaluating Model DSP

File name: GA_ModelDSP

Setting up the spreadsheet

When you open the file, you may receive a message similar to the one shown in *Figure C 1*.

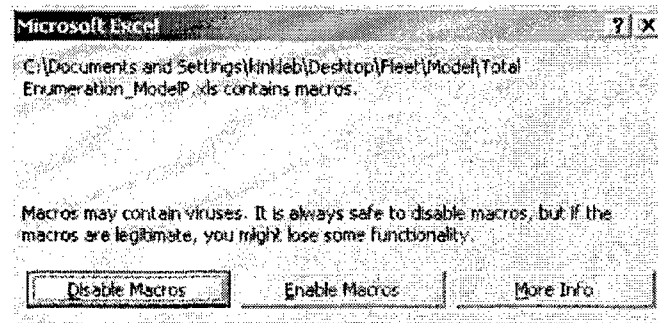


Figure C 1: Macro Notification

All the Visual Basic[®] code used to evaluate the model is written within a macro. Therefore, you should click **Enable Macros**. This will open the GA Settings worksheet with the default settings. Recall that population size is the number of candidate solutions or chromosomes that the GA monitors, and the number of generations is the number of times the population undergoes crossover and mutation. Finally, the seed value relates to the random number generator. This cell may have a value from 1 to 100. This worksheet is shown in *Figure C 2*.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Population Size	32										
3	Generations	1000										
4	Seed Value	2										
5												
6												
7												

Figure C 2: GA Settings Worksheet

Next, you must enter the parameters of the scenario you wish to evaluate. This is accomplished by entering data into the Inputs Worksheet. Fill in the number of systems, q , and subsystems, m , in the scenario you wish to evaluate. Then, determine the number of limited maintenance resources, s . Once you have entered these values, activate a blank cell (by clicking it), and then click **Reset Fields**.

Entering model parameter values

For each scenario you wish to evaluate, you must enter the model parameters. The parameters of interest are the failure rates of components in each subsystem, λ_j , the number of independent and identical copies of components in each subsystem, n_j , the amount of resource consumed by repairing a type j component, α_{jt} , and the amount of each resource available in each time bucket, β_t . These values must be entered for each of the s limited maintenance resources (in this case, $s = 2$) as shown in *Figure C 3*.

Consider the numerical example from *Section 5*. Recall that $q = 2$, $m = 3$, and $s = 2$, and note that $\beta_1 = 8$ and $\beta_2 = 6$. The next mission durations (in hours) are $t_1 = 4$, $t_2 = 1.75$, $t_3 = 2$, $t_4 = 0.5$, and the remaining model parameters are shown in *Table C 1*. *Figure C 3* provides a snapshot of the Inputs worksheet with the appropriate parameter inputs.

Table C 1: Model Parameters for Example Set of Systems

Subsystem	n_j	λ_j (in failures per hour)	α_1	α_2
1	2	0.10	2	3
2	3	0.16	3	1
3	2	0.06	2	2

Microsoft Excel - GA_ModelDSP.xls

File Edit View Insert Format Tools Data Window Help

Type a question for help

Arial 10 B I U

65%

Q47

195.84375 run time

Reset Fields

New Go

Instructions

Step 1 Enter g, m, and s values

Step 2 Press "Reset Fields"

Step 3 Enter data into input field

Step 4 Press "Go"

Number of Systems g 4

Number of Subsystems m 3

Number of limited resources s 2

Number of Failed components in each subsystem

1 2 3

1 2 3 4

Subsystem

1 2 3

0.10 2

0.16 3

0.06 2

length of each mission - ti

1 4

2 1.75 in minutes

3 2

4 3

Amount of resource consumed to repair a component within a subsystem - ai

1 2 3

2 3 2

3 1 2

Amount of each resource available - bi

1 8

2 6

Return, Start, and length of mission Times

1 2 3 4

GA Settings Inputs Output Print Out

Ready

NUM

Figure C 3: Inputs Worksheet with Example Parameters

Running an experiment

Once you have entered all the model parameters, you are now ready to run an experiment. Determine the number of failed components in each subsystem. If we again consider the numerical example from *Section 5*, the failed components as well as the return times for current mission and start times for future missions are shown in *Figure C 4*. These values are entered into the spreadsheet as shown in *Figure C 5*.

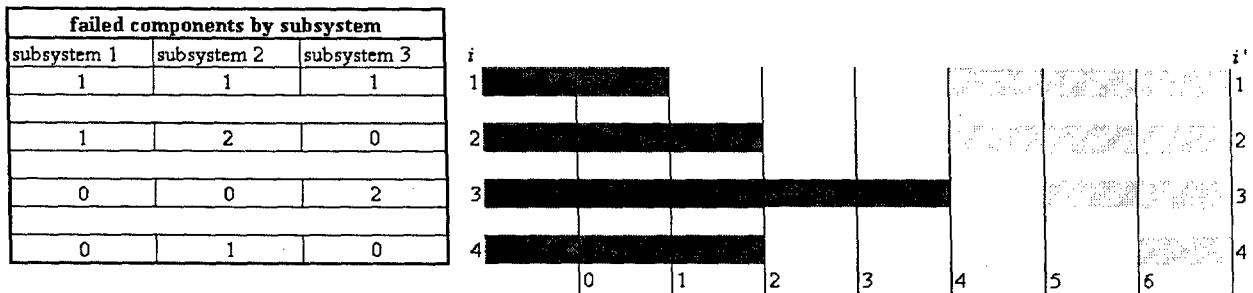


Figure C 4: Parameters for Dynamic Scenario Example

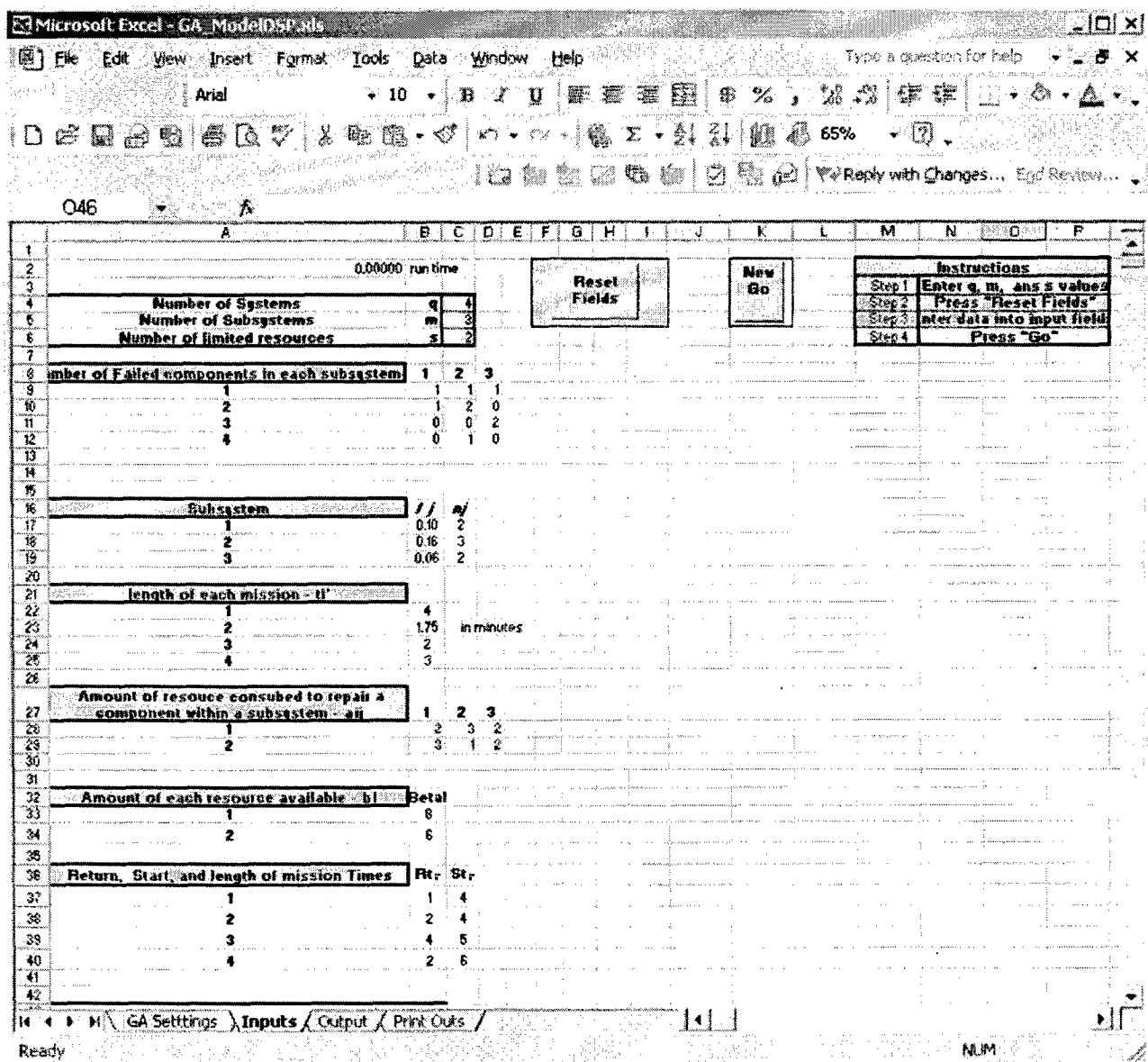


Figure C 5: Inputs Worksheet with All Example Parameters

To run the model, simply activate a blank cell and click **New Go**. The solution for the GA is shown in the Output worksheet. For this scenario, the GA ran for 195 seconds. The suggested maintenance actions and mission assignments are shown in *Figure C 6*.

Microsoft Excel - GA_ModelDSP.xls

File Edit View Insert Format Tools Data Window Help

Times New Roman 12 B I U

C24

	A	B	C	D	E	F	G	H	I
1									
2	Reliability	0.522484	i	j	h	dijh	Assign System 1 to mission 2		
3			1	1	1	1	Assign System 2 to mission 3		
4			1	3	2	1	Assign System 3 to mission 4		
5			2	1	3	1	Assign System 4 to mission 1		
6			2	2	4	1			
7			3	3	4	1			
8			3	3	5	1			
9			4	2	2	1			
10									
11									

GA Settings Inputs Output Print Outs

Ready

NUM

Figure C 6: Suggested Maintenance Actions and Mission Assignment